IACR Transactions on Symmetric Cryptology
ISSN 2519-173X, Vol. 2019, No. 4, pp. 23–53. DOI:10.13154/tosc.v2019.i4.23-53

Efficient Side-Channel Secure Message Authentication with Better Bounds

Chun Guo[1,2,3], François-Xavier Standaert[3], Weijia Wang[1,2,3] and Yu Yu[4,5]

[1] Key Laboratory of Cryptologic Technology and Information Security of Ministry of Education,
Shandong University, Qingdao, Shandong, 266237, China,
chun.guo@sdu.edu.cn,wjwang@sdu.edu.cn
[2] School of Cyber Science and Technology, Shandong University, Qingdao, Shandong, China
[3] ICTEAM/ELEN/Crypto Group, University of Louvain, Louvain-la-Neuve, Belgium
francois-xavier.standaert@uclouvain.be
[4] Computer Science and Engineering, Shanghai Jiao Tong University, Shanghai, China
yuyu@yuyu.hk
[5] State Key Laboratory of Cryptology, P.O. Box 5159, Beijing 100878, China

Abstract. We investigate constructing message authentication schemes from symmetric cryptographic primitives, with the goal of achieving security when *most intermediate values during tag computation and verification are leaked* (i.e., mode-level leakage-resilience). Existing efficient proposals typically follow the plain Hash-then-MAC paradigm $T = \mathsf{TGen}_K(\mathsf{H}(M))$. When the domain of the MAC function TGen_K is $\{0,1\}^{128}$, e.g., when instantiated with the AES, forgery is possible within time 2^{64} and data complexity 1. To dismiss such cheap attacks, we propose two modes: LRW1-based Hash-then-MAC (LRWHM) that is built upon the LRW1 tweakable blockcipher of Liskov, Rivest, and Wagner, and Rekeying Hash-then-MAC (RHM) that employs internal rekeying. Built upon secure AES implementations, LRWHM is provably secure up to (beyond-birthday) $2^{78.3}$ time complexity, while RHM is provably secure up to 2^{121} time. Thus in practice, their main security threat is expected to be side-channel key recovery attacks against the AES implementations. Finally, we benchmark the performance of instances of our modes based on the AES and SHA3 and confirm their efficiency.

Keywords: Message authentication · MAC · side-channel security · Hash-then-MAC · beyond-birthday-bound

1 Introduction

Message authentication (MA) schemes are fundamental symmetric primitives. A MA scheme allows two parties sharing a secret key K to authenticate data they send to each other. The sender applies a tag generation algorithm TGen to K and the message M to get a tag T, and then sends M, T to the receiver. The latter applies a verification algorithm Vrfy to K, a received message M, and its accompanying tag T, to get an output of 1 (accept) or 0 (reject), indicating whether or not the message should be considered authentic. There have been extensive studies on designing secure MA schemes with high performance. Most of them are based on conceptually simple primitives such as (tweakable) blockciphers and hash functions and enjoy provable security guarantees, e.g., CBC-MAC [BKR00], HMAC [BCK96], PMAC [BR02], and Wegman-Carter type MACs [WC81, CS16, DDNY18] to name a few. The simplest option for Vrfy is "Tag-then-compare", i.e., "If $\mathsf{TGen}(M) = T$ then return 1 else return 0". Such MA schemes

Received: 2019-06-01, Revised: 2019-09-01, Accepted: 2019-11-01, Published: 2020-01-31

are called *message authentication codes (MACs)* according to Bellare, Goldreich, and Mityagin [BGM04]. In fact, all the aforementioned MA schemes fall into this category.

Once deployed, the above "provably secure" MACs face a real-world threat that wasn't reflected in their classical proof models, i.e., *side-channel attacks* that utilize the information leakage of secrets in the deployed cryptosystems. Virtually all systems have specific types of leakages: for example, IoT, RFID, embedded systems, and smartphones allow measuring features (timing, power, etc) of physical actions during the algorithm executions [Koc96, KJJ99, CRR03] (also see [WYS$^+$18, RSWO17] for some recent practical examples), while computer systems may leak sensitive values due to memory corruptions [SPWS13] and malicious firmware [ZS18]. As a notable example, Ronen et al. [RSWO17] employed correlated power analysis to recover the key of AES-CCM that Philips uses to encrypt and authenticate firmware. This allows them to break integrity (of the CBC-MAC underlying CCM) and update worms into Philips Hue smart lamps, and the latter could further infect all such lamps via the IoT. It's thus not a surprise that side-channel security (with low overhead) has been explicit in the NIST requirements for authenticated encryption [oSN18].

In response to side-channel threats, the traditional countermeasures are applying implementation-level protections such as masking, shuffling, and hiding (see [MOP07]) to reduce the exploitable leakages from the implementations. But they induce significant overheads. For example, according to the recent results [GR17], the cycle counts of the (optimized) masked software implementations of blockciphers blows up by factors ranging from tenths to hundreds for number of shares ranging from 2 or 3 to more than 4, compared to a non-protected implementation. Significant overheads have also been observed in hardware as the number of shares increase [GMK17]. In addition, note that integrity of MA schemes usually relies on the secrecy of many other intermediate values, and infeasibility of side-channel key recovery isn't enough. For example, in CBC, one has to protect the XORs from side-channels to ensure the secrecy of the internal values, which is indeed crucial [DS09]. These additional protections further balloon the overhead.

To achieve better efficiency, the design of MA schemes with built-in side-channel security, a.k.a. *leakage-resilience* [DP08], has recently attracted wide attention. Unlike the "old school" MACs, leakage-resilient MA schemes are provably secure *in the presence of a reasonable amount of leakages*, thus enjoying a much stronger side-channel security. After some initial attempts [HLWW16, Sch10] (which are more or less theoretical due to unrealistic leakage assumptions or efficiency issues), the community eventually witnessed some practical designs that resist TGen leakages [PSV15, MOSW15, DEM$^+$17][1] and later some refined ones [BPPS17, BMOS17] that resist both TGen and Vrfy leakages.

Interestingly, while using very different premises, most of these practical designs, i.e., [MOSW15, DEM$^+$17, BPPS17, BMOS17], fall into the same (deterministic and stateless) hash-then-MAC (HtM) paradigm $T = \mathsf{FILTG}_K(\mathsf{H}(M))$, where H is a *keyless* hash function and FILTG_K is a somewhat side-channel secure Fixed Input-Length (FIL) tag generation function. The underlying reason is that Vrfy algorithms are inherently deterministic, and thus resisting Vrfy leakages essentially requires leakage-resilience in deterministic designs. As such, all keyed components require strong protection/leakage assumptions. In this respect, the use of keyless hash significantly mitigates the burden since it has no secret to protect: only the FIL MAC function FILTG_K needs to resist side-channel key recovery.

Moreover, $\mathsf{Vrfy}_K(M, T)$ shall be designed carefully. As stressed in [MOSW15, BMOS17, BKP$^+$18], the popular "Tag-then-compare" approach, i.e., first compute the "correct" tag $T_c = \mathsf{TGen}_K(M)$ and then compare if $T = T_c$, may leak information about T_c and degrade security when *Vrfy is susceptible to leakages.* The consensus of these works was that the designs shall avoid computing (and leaking) the "correct" tag T_c. For this, different methods have been employed: Berti et al. [BPPS17] used an invertible (side-channel secure)

[1] We remark that resistance to verification leakages was advertised in [DEM$^+$17], but the design IsapMac relies on carefully protecting the tag computations and comparisons—see our discussion later.

blockcipher E for the function FILTG, and compare if $\mathsf{H}(M) = \mathsf{E}_K^{-1}(T)$, while Barwell et al. [BMOS17] used a bilinear map for FILTG, and only operate on the (uncombined) pairings of the "correct" tag in their Vrfy implementation.

Birthday Bound Issue. The security of HtM is capped at the "birthday-bound": if the domain of FILTG is $\{0,1\}^n$, then a hash collision $\mathsf{H}(M) = \mathsf{H}(M')$ is found in $2^{n/2}$ time, after which $T = \mathsf{TGen}_K(M)$ gives rise to a forgery (M', T). When standard 128-bit blockciphers are used for FILTG, this attack with time 2^{64} and data complexity 1 raises a serious concern ($2^{63.1}$ computations *are* feasible [SBK$^+$17]). It's even worse in constrained environments with lightweight 64-bit blockciphers. Even more severely, when multiple instances of HtM with different keys are run by multiple users, a single collision $\mathsf{H}(M) = \mathsf{H}(M')$ allows forgery for every instance.

Theoretically, the issue can be resolved by simply using a FILTG with large domain. But this solution isn't optimal, as the lack of reliable 256-bit blockciphers make it hard to instantiate. Besides, other instantiations, e.g., pairing [MOSW15], typically result in large performance penalty. One may also resort to protected implementations of MACs (e.g., EWCDM [CS16, MN17]) with black-box beyond-birthday-bound (BBB) security, but the cost would be much higher than dedicated leakage-resilient schemes.

Pereira et al.'s nonce-based HMAC-like MAC doesn't suffer from such low-data attacks, but it is vulnerable to Vrfy leakages [PSV15] (moreover, its provable bound remains birthday). Such a limitation can be overcame by using other primitives. For example, using a tweakable blockcipher (TBC) $\widetilde{\mathsf{E}}$, Berti et al. proved that the Hash-then-TBC MA scheme, i.e., $\mathsf{TGen}_K(M) = \widetilde{\mathsf{E}}_K^V(U)$, $U\|V = \mathsf{H}(M)$, is secure up to BBB $2^n/n$ queries [BGP$^+$19].[2]

Alternatively, Dobraunig et al.'s sponge-based IsapMac is an HtM instance with a sponge-based hash for H and a duplex-based function for FILTG [DEM$^+$17]. The rate of the duplex is 1, so that every secret state value can be involved in only two possible permutation-calls, and this renders DPAs infeasible by design. The domain of their FILTG is larger than $\{0,1\}^{128}$, thus IsapMac has 128-bit security. But the Vrfy of IsapMac has to be "Tag-then-compare" as its FILTG is not invertible, and this reduces its mode-level leakage-resilience. Indeed, having an invertible primitive is instrumental in providing strong integrity guarantees in the work by Berti et al., and an IsapMac implementation should therefore have its actions involving the right tag T_c protected against side-channels.[3]

While the efficiency of these solutions seems optimal in terms of the number of side-channel protected executions, in many cases, e.g., when using the OpenSSL[4] or the HACL crypto libs [ZBPB17], TBCs and Sponges (in side-channel protected forms) are not (yet) readily available. Based on the above, it appears that the design of side-channel secure MA schemes with BBB security in the presence of verification leakage using simple symmetric primitives such as blockciphers and keyless hash functions is still missing in the current state-of-the-art. We therefore focus on this challenge, which we believe is particularly relevant to take advantage of standards for which implementations are well understood and readily available. We stress that designing blockcipher-based deterministic BBB-secure MACs is already believed challenging *even without leakage* [PS15]. The presence of verification leakages as well as our attempt to minimize the number of calls to (protected) keyed blockciphers undoubtedly amplify the difficulty.

Our Contributions: Modes with BBB Leakage MAC Security. We propose two new MA modes LRWHM and RHM that make a call to a $2n$-bit hash function and two calls to a side-channel protected blockcipher. They can be seen as instances of the aforementioned Hash-then-TBC paradigm, with the TBC instantiated via a classical blockcipher.

[2] Earlier, List and Nandi defined Hash-then-TBC paradigm based on *keyed* hash functions [LN17]. Their motivation is quite different from ours (reducing the amount of leakages).

[3] Whether protecting the tag computations (and comparisons) or using mode-level leakage-resilience is the most efficient option remains an implementation-dependent open question.

[4] `https://www.openssl.org/docs/OpenSSL300Design.html`.

In detail, we first consider instantiating the TBC with the LRW1 tweakable blockcipher $\widetilde{\mathsf{E}}(tw, x) = \mathsf{E}_{K_2}(tw \oplus \mathsf{E}_{K_1}(x))$ of Liskov et al. [LRW11],[5] and obtain our first mode LRW1-based Hash-then-MAC ($\mathsf{LRWHM}^{\mathsf{H,E}}$) depicted in Fig. 1 (left). To analyze its leakage security, we assume that the blockcipher E is "completely" secure against side-channel key recovery, via modeling E as a "normal" strong PRP that leaks nothing (i.e., we model it as a "leak-free" component). This model abstracts the details within the implementation of E and nicely allows us to focus on arguing the harmlessness of *mode-level leakages*. For the latter, we make a very conservative assumption that *all intermediate values appearing during tag generation and verification are leaked in full* (jumping ahead, see $\mathsf{L}_{\mathsf{TGen}}$ and $\mathsf{L}_{\mathsf{Vrfy}}$ in Fig. 2). This setting, also used in [BKP+18, BPPS17, BGP+19], is a special case of the *continuous leakage model* since the total amount of leakage continuously increases during the lifetime of the system. While appearing theoretical, this assumption enables deriving simple security *lower bounds* against mode-level leakages. On the other hand, we model the hash function H as a $2n$-bit random oracle RO. With the above assumptions, we prove that $\mathsf{LRWHM}^{\mathsf{H,E}}$ is unforgeable against leakage up to BBB $2^{2n/3}/n$ queries.

Given the literature, it appears quite difficult to beat the $2^{2n/3}$ bound with two blockcipher-calls. In order to achieve higher security, we instantiate the TBC with a rekeying-per-input function that bears some resemblance to a TBC of Minematsu [Min09], and this results in our second mode Rekeying Hash-then-MAC $\mathsf{RHM}^{\mathsf{H,E}}$ as shown in Fig. 1 (right). This mode works with an n-bit key that is then used to generate a fresh key for the second cipher. It is known that BBB proofs for such rekeying-based modes have to rely on the Ideal Cipher Model (ICM) [BKR98, ST16, Men17]. Therefore, we follow this and prove $\mathsf{RHM}^{\mathsf{H,E}}$ unforgeable against leakage up to asymptotically optimal $2^n/n$ queries.

In summary, LRWHM is closer to the "standard" solution to BBB secure MACs (i.e., without the slightly inefficient rekeying and the ICM), and ensures a standard BBB $2^{2n/3}/n$ security. RHM is appealing in practice as it achieves optimal bounds (thus more secure against "brute force" attacks) within two blockcipher-calls. Yet, the security insurance of RHM, proved in the ICM, turn a bit heuristic once instantiated [CGH98].

Practical Issues and Comparison. Our modes can be instantiated with AES128 and SHA3, resulting in concrete instances AES-SHA3-LRWHM and AES-SHA3-RHM: both produce 128-bit tags, while the key size of the former is 256-bit which is larger than the latter's 128-bit. It's natural to ask what sort of security is guaranteed on these deployed instances. For this, note that our provable bounds indicate that as long as the AES keys remain "safe", AES-SHA3-LRWHM is secure up to $2^{78.3}$ time complexity, while AES-SHA3-RHM is secure up to 2^{121} time. Such computations can be seen as impractical. Therefore, *in practice the main threat to these instances is expected to be the side-channel key recovery attack against the AES implementations.* The concrete side-channel security depends on the implementations that can't be "predicated" in advance, but the data complexity could be $> 2^{20}$ in some cases [GMK17] and would further increase with the level of protections. This is clearly more expensive than the aforementioned low-data attack against the plain HtM scheme, i.e., we've achieved our goal.

In theory, deploying classical MACs in leaking scenarios necessarily requires protecting all the operations. Upon a message with ℓ blocks (in the corresponding sense), this typically consumes $\geq \ell$ executions of some heavy protected circuits: for example, $\geq \ell$ blockcipher-calls in CBC and PMAC [Rog04], $\geq \ell$ large field multiplications in universal hash-based MACs (like Wegman-Carter [WC81]), $\geq \ell$ compression function calls in HMAC, and $\geq \ell$ permutation-calls in (the SHA3-based) KMAC [KCP16]. In contrast, in our modes (and all the other HtM-based modes), such a message triggers $O(\ell)$ calls to a light *unprotected* hash circuits and 1 or 2 calls to some heavy side-channel secure primitive. The performance gain is obvious. To be more convincing, we implement and estimate the performances. Concretely, we reuse the C code of Dobraunig et al. [DEM+19] to ease a comparison

[5] Two TBC constructions were proposed in [LRW11] and subsequently named LRW1 and LRW2 [LST12].

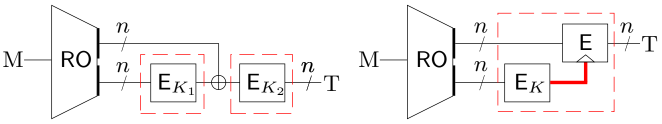

Figure 1: Our new MA modes. The hash H is modeled as a random oracle RO. Components in the red dashed squares are protected to be side-channel secure (modeled as "leak free"): these include all the 4 blockcipher calls and the red bold wire in the right part. (left) LRW1-based hash-then-MAC LRWHM; (right) rekeying hash-then-MAC RHM.

to the IsapMac variants underlying Isap-K-128 and Isap-K-128A (which we denote IsapMacK and IsapMacKA resp.). This code contains a Keccak-f[400] permutation, upon which we build a SHA3 variant for our modes (this means our hash is very close to IsapMacKA rather than the standard SHA3). For masked AES, we implement the proposals of [GR17]. Using these components, we implement AES-SHA3-RHM, AES-SHA3-LRWHM, IsapMacK, IsapMacKA (as they are the only known leakage-resilient MACs with both efficiency and concrete security comparable to ours),[6] and a variant of AES-CBC (as a representative of the "fully protected" classical MACs), and compare their performances. The conclusions are: (a) ours outperform IsapMacK when the level of side-channel protections is not too strong (i.e., less than 10th order masking), and (b) ours are comparable to the more aggressive variant IsapMacKA, and (c) ours outperform the protected AES-CBC as long as the messages are not too short (e.g., more than 50 bytes). See Section 4 for details.

For clearness, we serve a comprehensive comparison between our new proposals and the IsapMacs. Indeed we have quite different design goals. Regarding practicality, we emphasized more on easy deployment from any crypto lib (e.g., the HACL lib [ZBPB17]) that has implemented hashing and masked blockciphers (with these, even ones very unfamiliar with implementations could deploy our schemes using several lines of codes), and we enable a modular approach, allowing updating the primitives for better security or switching to more masking-friendly ciphers [GLSV15] for better efficiency. On the other hand, IsapMacs offer more dedicated designs (somewhat supported by the recent leakage-resilience proofs [DM19, GPPS19] for duplex) aiming at (potentially) better efficiency. Regarding security, we aim at high side-channel security guarantees (at the cost of some necessary expertise since masking is non-trivial to implement), and our modes "inherit" the security of the (well-understood) primitives in use, whereas IsapMacs aim to trade a bit of these high security guarantees for a scheme enabling the default implementation to provide built-in side-channel resistance. This tradeoff is in particular visible in the case of decryption leakages where repeated measurements may allow obtaining noise-free leakage traces for IsapMacs (raising the risk of advanced algebraic/analytical attacks [RSV09, VGS14]) while masking should mitigate this risk in our schemes.

Potential Applications. In side-channel sensitive settings, our MA modes could be deployed "alone" for sole integrity, or used for improving authenticated encryption (AE) schemes [BN08]. For example, the aforementioned Philips smart lamps used AES-CCM [WHF], which is a sophisticated combination of AES-CTR and AES-CBC MAC. To foil the (forgery-based) attack of [RSWO17], one could in principle protect the AES-CBC implementation. But an alternative approach is to replace AES-CBC by a protected implementation of AES-SHA3-RHM and use a standard Encrypt-then-MAC composition AES-SHA3-CTR-then-RHM, which we abbreviate as CRHM. A bit more concretely, CRHM

[6] We omit the estimation of the pairing-based HtM of [MOSW15, BMOS17], as it appears rather inefficient, consuming ≈ 4 seconds to produce a tag on 32-bit ARM Cortex-M4 CPU [MOSW15]. The advantage of [MOSW15, BMOS17] is provable security against adaptively-chosen leakage functions.

uses two keys K_1 and K_2 for the 1st pass CTR and 2nd pass RHM, and the 2nd pass RHM produces a tag based on the nonce N, the associated data A and the ciphertext C of CTR. Taking N and A as the inputs of RHM prevents trivial forgeries. The black-box AE security (with no leakage) of this composition follows from the security of CTR, the strong unforgeability of RHM (implied by our strong unforgeability result with leakages), and the standard result on Encrypt-then-MAC composition [BN08, Theorem 4.4]. With leakages, this AE still offers a high security against forgery attacks thanks to RHM. As for performance, it achieves much lower latency and energy consumption according to our evaluation, as long as N, A, and M/C accumulate to more than 50 bytes.

We remark that while applications such as IoT were believed to mostly transfer short messages, some technologies do allow larger packets (e.g. up to 243 bytes for LoRa [RKS17]). Moreover, in the CCA setting the scheme has to handle long inputs: this could happen in the DDoS attack scenario, in which the adversary could send many invalid long ciphertexts to trigger verification and cause a huge resource consumption. Therefore, the ability of efficiently handling moderately long messages remains of importance for IoT and similar settings.

On the other hand, if AES-SHA3-HtM is used, leading to CTR-then-HtM, then the fatal term $t^2/2^{128}$ remains in the (standard model) provable bound due to the hash collision (t denotes the time complexity. See, e.g., [BPPS17, Theorem 4]), rendering it less reliable.

It's worth noting that the application of BBB secure HtM variant may be far beyond side-channel security. For example, in [RSS17], CTR-then-HtM was identified as the most efficient AE suitable for evaluation in multi-party computation engines. The performance advantage of the HtM paradigm stems from the fact that keys are held in secret shared form in this setting, and thus paradigms with a minimal number of calls to keyed primitives outperform the others. Clearly, the composition CTR-then-RHM could also be used here to dismiss the fatal term $t^2/2^{128}$ caused by the plain HtM.

Further Related Work. Various MACs with BBB black-box security have been proposed: DWCDM [DDNY18], 3kf9 [ZWSW12], PMAC variants [Yas11, DDN+17], Double-block Hash-then-Sum [DDNP18], and those from TBCs [IMPS17, CLS17] and compression functions [Yas09]. We quote a MAC [DS11] that was also proved BBB secure under the "unbounded leakage" assumption. This design is rather complicated and consumes $\ell \cdot poly(n)$ number of side-channel protected blockcipher-calls upon ℓ-block messages. This construction is more attractive from a theoretical point of view, as it's a *BBB secure MAC domain extender*. We rather provide modes that are simple and easy to deploy.

There are quite a number of leakage-resilient modes that could (potentially) be instantiated with blockciphers, including pseudorandom number generators (PRNGs) [DP08, PSP+08, Pie09, YSPY10, SPY13] and (authenticated) encryption [PSV15, BKP+18, BPPS17]. Though, the PRNGs only rely on leaky blockcipher executions which differs from the "leak-free" blockciphers in this paper. The latter model was first used in [PSV15].

Finally, we remark that our proofs seem more complicated than the black-box MAC proofs, and bear some resemblance to *indifferentiability* [MRH04] proofs, e.g., to [ABD+13].

Organization. We serve notations and definitions in Section 2. Then, in Section 3 we present the two new modes and their security proofs. In Section 4 we benchmark the performances of their concrete instances and make comparison.

2 Preliminaries

General Notations and Definitions. For a finite set $\mathcal{X}$, $X \overset{\$}{\leftarrow} \mathcal{X}$ denotes selecting an element from $\mathcal{X}$ uniformly at random and $|\mathcal{X}|$ denotes its cardinality. In all the following, we fix an integer $n \geq 1$. Further denote by $\mathcal{H}(2n)$ the set of all functions of domain $\{0,1\}^*$ and range $\{0,1\}^{2n}$, by $\mathcal{P}(n)$ the set of all permutations on $\{0,1\}^n$, and by $\mathcal{BC}(\kappa, n)$ the set of all blockciphers with n-bit block-size and κ-bit keys (though, we will mainly use $\kappa = n$ in this paper). Finally, for $U, X \in \{0,1\}^n$, $U \| X$ or simply UX denotes their concatenation.

ADVERSARY. We denote by a (q,t)-adversary a probabilistic algorithm that has access to several oracles (the number of which depends on the concrete context), can make at most q queries to its (multiple) oracles, and can perform computation bounded by running time t.

STRONG PSEUDORANDOM PERMUTATION. For the security analysis of LRWHM, we model the underlying blockcipher as a strong pseudorandom permutation, abbreviated as SPRP. Formally, for an n-bit blockcipher $\mathsf{E} : \{0,1\}^\kappa \times \{0,1\}^n \to \{0,1\}^n$, the SPRP advantage of a (q,t)-adversary $\mathcal{A}$ is

$$\mathbf{Adv}^{\mathsf{SPRP}}_{\mathsf{E}}(\mathcal{A}) := \left| \Pr[k \xleftarrow{\$} \{0,1\}^\kappa : \mathcal{A}^{\mathsf{E}_k, \mathsf{E}_k^{-1}} = 1] - \Pr[\mathsf{P} \xleftarrow{\$} \mathcal{P}(n) : \mathcal{A}^{\mathsf{P}, \mathsf{P}^{-1}} = 1] \right|.$$

IDEAL PRIMITIVES. A hash function $\mathsf{RO} : \{0,1\}^* \to \{0,1\}^{2n}$ which is sampled uniformly at random from the set $\mathcal{H}(2n)$ is called a *random oracle*. Similarly, a blockcipher E : $\{0,1\}^\kappa \times \{0,1\}^n \to \{0,1\}^n$ sampled uniformly at random from $\mathcal{BC}(\kappa, n)$ is called an *ideal cipher*. In this case E_K is a random permutation of $\{0,1\}^n$ for each $K \in \{0,1\}^n$.

Message Authentication and Its Leakage Security. Following Bellare et al. [BGM04], a Message Authentication (MA) scheme is a tuple of two polynomial time algorithms $\mathsf{Scheme} = (\mathsf{TGen}, \mathsf{Vrfy})$ defined as follows:

- the **tag generation** algorithm $\mathsf{TGen}_K(M)$ takes as input the secret key K and the message M, and then outputs a tag T;
- the deterministic, stateless **verification** algorithm $\mathsf{Vrfy}_K(M,T)$ takes as input the secret key K, a message M, and a tag T. The algorithm outputs 1 (*accept*) if the tag is valid for the message, else it outputs 0 (*reject*).

Informally, the MA scheme Scheme is said to be *strongly unforgeable* [BN08], if the adversary is unsuccessful in the following security game. First, a key K is selected as part of the experiment. Next, the adversary $\mathcal{A}$ can arbitrarily choose messages and ask for tags under the key K, or ask for verify the correctness of a message-tag pair (also under the key K). Following [CS16], the adversary is *non-trivial*, in the sense that it never asks a verification query $\mathsf{Vrfy}(M,T)$ if a previous tagging query $\mathsf{TGen}(M)$ returned T. Under this restriction, we say that $\mathcal{A}$ forges if any of its queries to Vrfy returns 1 (accept).

We denote by LTGen and LVrfy the leaking implementation of TGen and Vrfy algorithms resp. LTGen runs both TGen and a *leakage function* $\mathsf{L}_{\mathsf{TGen}}$ which captures the additional information given by an implementation of TGen during its execution, and returns the outputs of both TGen and $\mathsf{L}_{\mathsf{TGen}}$ which all take the same input; similarly for LVrfy and $\mathsf{L}_{\mathsf{Vrfy}}$. Later in Section 3, we will explicitly define $\mathsf{L}_{\mathsf{TGen}}$ and $\mathsf{L}_{\mathsf{Vrfy}}$ for each MA mode. The rules are:

1. A blockcipher-call $\mathsf{E}_K(X) \to Y$ leaks X and Y but never K;
2. An inverse blockcipher-call $\mathsf{E}_K^{-1}(Y) \to X$ leaks X and Y but never K;
3. An XOR action $X \oplus Y \to Z$ leaks all: X, Y, and Z;
4. A composed execution $\mathsf{E}_{\mathsf{E}_K(X)}(Y) \to Z$ leaks X, Y, and Z, but never K nor $\mathsf{E}_K(X)$;
5. A composed execution $\left(\mathsf{E}_{\mathsf{E}_K(X)}\right)^{-1}(Z) \to Y$ leaks X, Y, and Z, but never K nor $\mathsf{E}_K(X)$;
6. A hash-call $\mathsf{H}(M) \to U \| X$ leaks M, U, and X.

$\mathsf{TGen}_K^{\mathsf{H},\mathsf{E}}(M)$: proceeds in two steps:

1. **Parses** $K = K_1 \| K_2$ and **computes** $U \| X = \mathsf{H}(M)$, $V = \mathsf{E}_{K_1}(U)$, $Y = V \oplus X$, and $T = \mathsf{E}_{K_2}(Y)$;
2. **Return** T as the tag. According to the rules specified in Section 2, the leakage function is $\underline{\mathsf{L}_{\mathsf{TGen}}(K, M) := (U, V, X, Y, T).}$

$\mathsf{Vrfy}_K^{\mathsf{H},\mathsf{E}}(M, T)$: parses $K = K_1 \| K_2$, then:

1. *Forward Computation*: **computes** $U \| X = \mathsf{H}(M)$;
2. *Backward Computation*: $Y' = \mathsf{E}_{K_2}^{-1}(T)$, $V' = X \oplus Y'$, and $U' = \mathsf{E}_{K_1}^{-1}(V')$;
3. **If** $U = U'$, **then return** 1, **else return** 0. According to the rules specified in Section 2, the leakage function is $\underline{\mathsf{L}_{\mathsf{Vrfy}}(K, M, T) := (U, X, U', V', Y').}$

Figure 2: The description of $\mathsf{LRWHM}^{\mathsf{H},\mathsf{E}}$ MA mode.

During the MA security interaction, if either the tag generations or the verifications give corresponding leakage along with the answer, then we are in the *MA security definitions in the presence of leakage*. In contrast, we never consider leakages of the key generation, as the actual way of loading the key into a device can vary quite a lot from one situation to another, and will usually happen at manufacturing time, out of reach of the adversary. In this paper, we always consider the setting where *both* TGen oracle leakages *and* the Vrfy oracle leakages are presented. Formally, we define

$$\mathbf{Adv}_{\mathsf{Scheme}}^{\mathsf{MAL2}}(\mathcal{A}) := \Pr_K[\mathcal{A}^{\mathsf{RO},\mathsf{LTGen}_K,\mathsf{LVrfy}_K} \text{ forges}]. \tag{1}$$

The suffix 2 indicates that the number of involved leaking oracles is two; this follows the convention in [GPPS]. The presence of RO is in accordance with our use of the random oracle model.

3 MA Modes and Provable Results

In this section we present the formal definitions of our modes and their provable results. Interpretations of the theorems are deferred to the next section.

3.1 Mode LRWHM and Its Security

Formally, the mode $\mathsf{LRWHM}^{\mathsf{H},\mathsf{E}}$ along with the leakages is defined in Fig. 2. With the "leak-free" blockcipher plus "unbounded" mode-level leakage assumptions, the $\mathsf{MAL2}$ security of $\mathsf{LRWHM}^{\mathsf{H},\mathsf{E}}$ is proved up to $2^{2n/3}/n$ queries.

Theorem 1. *Assume $2^n \geq 8$ and $4 \leq q \leq 2^n/2$. Let $\mathsf{E} : \{0,1\}^\kappa \times \{0,1\}^n \to \{0,1\}^n$ be a blockcipher and $\mathsf{RO} : \{0,1\}^* \to \{0,1\}^{2n}$ be a random oracle. Then with the leakages specified by $\mathsf{L}_{\mathsf{TGen}}$ and $\mathsf{L}_{\mathsf{Vrfy}}$ (Fig. 2), for any (q,t)-adversary $\mathcal{A}$ against the $\mathsf{MAL2}$ security of $\mathsf{LRWHM}^{\mathsf{RO},\mathsf{E}}$, there exists a $(q, t + q \cdot t_{\mathsf{LRWHM}})$-adversary $\mathcal{A}'$ against the SPRP security of E, such that t_{LRWHM} stands for the time to evaluate $\mathsf{LRWHM}^{\mathsf{RO},\mathsf{E}}$ once (with an adversary-picked key), and that*

$$\mathbf{Adv}_{\mathsf{LRWHM}^{\mathsf{RO},\mathsf{E}}}^{\mathsf{MAL2}}(\mathcal{A}) \leq 2\mathbf{Adv}_{\mathsf{E}}^{\mathsf{SPRP}}(\mathcal{A}') + \frac{32nq^3}{2^{2n}} + \frac{70q^{3/2}}{2^n}. \tag{2}$$

Proof. While the mode is simple, its analysis has to be dedicated and quite non-trivial, since Liskov et al. only proved $2^{n/2}$ security for LRW1 TBC [LRW11] and do not support the modular way. To ease understanding, below we first overview the proof ideas and steps in subsubsection 3.1.1, and then present the main steps in subsubsections 3.1.2 and 3.1.3.

3.1.1 Proof Overview

Based on the adversarial power, we make some initial observations:

(i) During the interaction, RO is queried at most q times, since each such query is either directly made by $\mathcal{A}$, or made by TGen or Vrfy which is ultimately made by $\mathcal{A}$;

(ii) similarly, the number of calls to E_{K_1}, resp. E_{K_2}, is at most q.

Recall that our goal is to bound

$$\mathbf{Adv}^{\mathsf{MAL2}}_{\mathsf{LRWHM}^{\mathsf{RO,E}}}(\mathcal{A}) = \Pr[K \xleftarrow{\$} \{0,1\}^{2\kappa}, \mathsf{RO} \xleftarrow{\$} \mathcal{H}(2n) : \mathcal{A}^{\mathsf{RO},\mathsf{LTGen}^{\mathsf{RO,E}}_K,\mathsf{LVrfy}^{\mathsf{RO,E}}_K} \text{ forges}].$$

The first step, idealizing the scheme, is standard for MAC security proofs. In detail, we replace the calls to E_{K_1} and E_{K_2} underlying $\mathsf{LTGen}^{\mathsf{RO,E}}_{K_1\|K_2}$ and $\mathsf{LVrfy}^{\mathsf{RO,E}}_{K_1\|K_2}$ by two independent random permutations $\mathsf{P} = (\mathsf{P}_1, \mathsf{P}_2)$, and denote by $\mathsf{LTGen}^{\mathsf{RO,P}}$ and $\mathsf{LVrfy}^{\mathsf{RO,P}}$ the obtained idealized oracles. By a straightforward hybrid argument,[7] there is an adversary $\mathcal{A}'$ that makes at most q oracle queries and evaluates (a certain part of) $\mathsf{LRWHM}^{\mathsf{RO,E}}$ for at most q times (thus the running time $t + q \cdot t_{\mathsf{LRWHM}}$), such that

$$\left|\mathbf{Adv}^{\mathsf{MAL2}}_{\mathsf{LRWHM}^{\mathsf{RO,E}}}(\mathcal{A}) - \mathbf{Adv}^{\mathsf{MAL2}}_{\mathsf{LRWHM}^{\mathsf{RO,P}}}(\mathcal{A})\right| \le 2\mathbf{Adv}^{\mathsf{SPRP}}_{\mathsf{E}}(\mathcal{A}'). \tag{3}$$

where

$$\mathbf{Adv}^{\mathsf{MAL2}}_{\mathsf{LRWHM}^{\mathsf{RO,P}}}(\mathcal{A}) = \Pr[\mathsf{P} \xleftarrow{\$} (\mathcal{P}(n))^2, \mathsf{RO} \xleftarrow{\$} \mathcal{H}(2n) : \mathcal{A}^{\mathsf{RO},\mathsf{LTGen}^{\mathsf{RO,P}},\mathsf{LVrfy}^{\mathsf{RO,P}}} \text{ forges}]$$

We then derive an upper bound for $\mathbf{Adv}^{\mathsf{MAL2}}_{\mathsf{LRWHM}^{\mathsf{RO,P}}}(\mathcal{A})$. The idea is that, if a non-trivial verification query returns 1, then right after this query returns, there exists a "chain" of historical query-records $\mathsf{RO}(M) = UX$, $\mathsf{P}_1(U) = V$, and $\mathsf{P}_2(V \oplus X) = T$ such that the tag generating action $\mathsf{LTGen}^{\mathsf{RO,P}}(M) \to T$ never happened before. It then suffices to prove that such chains are unlikely to occur. As will be clear in the proof, the presence of such a chain is typically due to unexpected collisions between the *three* query-records within the chain (for example, an RO query $\mathsf{RO}(M) = UX$ collide with $P_1(U') = V'$ and $P_2(Y') = T'$, in the sense that $U = U'$ and $X = V' \oplus Y'$), and thus the probability could go beyond the birthday. In addition, a crucial property is that verification queries are checked using the inverse, so that the involved permutation query-records have random "endpoints" at the input side and low probability to collide with existing values. It can be seen that, if verification is defined in the classical inverse-free manner, then a single verification query could create such a "chain" and leak it to $\mathcal{A}$ for forging, and no proof is possible.

To formalize the above ideas, we analyze the adversarial interaction with the idealized oracles $\mathsf{LTGen}^{\mathsf{RO,P}}$ and $\mathsf{LVrfy}^{\mathsf{RO,P}}$ using the game-playing technique [BR06]. We describe the security game in Fig. 3. The game offers three interfaces to $\mathcal{A}$ to mimic the oracles $\mathsf{LTGen}^{\mathsf{RO,P}}$, $\mathsf{LVrfy}^{\mathsf{RO,P}}$, and RO (captured by the statements following "When $\mathcal{A}$ asks..."). It also has 4 secret procedures P_1, P_1^{-1}, P_2, and P_2^{-1} for internal random permutation calls. To mimic the ideal oracles, the game maintains three sets ROSet, PSet_1, and PSet_2 for already defined RO, P_1, and P_2 query-records, and uses lazy sampling to gradually create new records. The game also maintains a global query counter *qnum* to indicate the timestamp of the records. Therefore, a record in the set ROSet is of the form (M, UX, num) with $M \in \{0,1\}^*, U, X \in \{0,1\}^n$, indicating the relation that $\mathsf{RO}(M) = UX$ and that

[7] This is possible due to our "leak-freeness" assumption on E. Obviously, "leak-freeness" assumption is a bit intuitive, and how to remedy the situation is an open question.

the record was created when $qnum = num$. A record in the set PSet_1 is of the form (U, V, dir, num) indicating similar meanings. The additional field dir indicates the direction of the internal P_1 query that produces this record: $dir = \rightarrow$ means it was a forward query $\mathsf{P}_1(U) \to V$, while $dir = \leftarrow$ means it was backward $\mathsf{P}_1^{-1}(V) \to U$. The set PSet_2 is just similar to PSet_1. In addition, four quantities α, β, γ_1, and γ_2 are used in Fig. 3, which are defined as

$$\begin{aligned}
\alpha &:= \big|\{((M, UX, \star), (U', V', \star, \star)) \in \mathrm{ROSet} \times \mathrm{PSet}_1 : U = U' \wedge M \notin \mathrm{TGened}\}\big|, && (4)\\
\beta &:= \max_{V \in \{0,1\}^n} \mu_V \\
&= \max_{V \in \{0,1\}^n} \big|\{((M, UX, \star), (Y, T, \star, \star)) \in \mathrm{ROSet} \times \mathrm{PSet}_2 : V = X \oplus Y\}\big|, && (5)\\
\gamma_1 &:= \big|\{((M, UX, \star), (M', U'X', \star)) \in \mathrm{ROSet}^2 : M \neq M' \text{ and } U = U'\}\big|, && (6)\\
\gamma_2 &:= \big|\{((M, UX, \star), (M', U'X', \star)) \in \mathrm{ROSet}^2 : M \neq M' \text{ and } V = V'\}\big|, && (7)
\end{aligned}$$

Finally, the game also maintains a set TGened for the messages involved in earlier $\mathsf{LTGen}^{\mathsf{RO},\mathsf{P}}$ queries, i.e., $M \in \mathrm{TGened}$ if and only if $\mathsf{LTGen}^{\mathsf{RO},\mathsf{P}}(M)$ has been queried.

As in typical game-based proofs, we specify several "bad events" that may lead to chains of records in future, and force the game to **abort** (as shown in Fig. 3) when any of the events occur. Once abortion occurs, we write "$\mathcal{A}^{\mathsf{RO},\mathsf{LTGen}^{\mathsf{RO},\mathsf{P}},\mathsf{LVrfy}^{\mathsf{RO},\mathsf{P}}}$ aborts". In the remaining, we proceed by first upper bounding $\Pr[\mathcal{A}^{\mathsf{RO},\mathsf{LTGen}^{\mathsf{RO},\mathsf{P}},\mathsf{LVrfy}^{\mathsf{RO},\mathsf{P}}}$ aborts$]$ in subsubsection 3.1.2, and then arguing that as long as abortion does not occur, chains of records are unlikely to occur—and thus $\mathcal{A}$ is unlikely to forge—in subsubsection 3.1.3.

3.1.2 Probability of Abortion

Here we devote to prove $\Pr[\mathcal{A}^{\mathsf{RO},\mathsf{LTGen}^{\mathsf{RO},\mathsf{P}},\mathsf{LVrfy}^{\mathsf{RO},\mathsf{P}}}$ aborts$] \leq 6q^{3/2}/2^n$. For this, we consider the conditions in Fig. 3 in turn. First, (B-1) essentially captures the event of collision within the RO queries, thus $\Pr[\text{(B-1)}] \leq \frac{q(q-1)}{2^{2n+1}}$.

We then consider (B-2). Its first half states that there exist 3 distinct queries $(M, UX, \star), (M', U'X', \star), (M'', U''X'', \star)$ such that $U = U' = U''$, the probability of which is $\leq \binom{q}{3} \cdot \frac{1}{2^{2n}}$. Analysis for the second half is similar. By the above,[8]

$$\Pr[\text{(B-1)} \vee \text{(B-2)}] \;\leq\; \frac{q(q-1)}{2^{2n+1}} + 2 \cdot \frac{q(q-1)(q-2)}{6 \cdot 2^{2n}} \;\leq\; \frac{q^3}{2^{2n+1}} \;\leq\; \frac{q^{3/2}}{2^n}. \tag{8}$$

For the condition (B-3), the quantity γ_1 is viewed as a random variable over the random choice of RO. Note that

$$\mathbb{E}[\gamma_1] \leq \sum_{(M,UX,\star) \neq (M',U'X',\star)} \Pr\big[U = U'\big] \leq \frac{q^2}{2^n}.$$

Using Markov inequality we obtain

$$\Pr[\text{(B-3)}] = \Pr[\gamma_1 \geq \sqrt{q}] \leq \frac{q^2}{2^n \sqrt{q}} = \frac{q^{3/2}}{2^n}. \tag{9}$$

[8] In Theorem 1, it may be tempting to separate different types of queries, i.e., to derive the bounds based on the assumption that $\mathcal{A}$ makes q_h, q_m, and q_v queries to the oracles RO, LTGen, and LVrfy resp., with the hope of getting ride of the data complexity-independent term $q_h^{3/2}/2^n$. But this is not successful: with the new notations, RO is queried at most $Q = q_h + q_m + q_v$ queries, and thus $\Pr[\text{(B-1)} \vee \text{(B-2)}]$ "just" changes to $Q^{3/2}/2^n$—the term $q_h^{3/2}/2^n$ remains. We thereby eschew this approach for simplicity. Whether the term $q_h^{3/2}/2^n$ can be avoided via improved analyses is an open question..

```
Initialization:
  ROSet := $\emptyset$, TGened := $\emptyset$, PSet$_1$ := $\emptyset$, PSet$_2$ := $\emptyset$, $qnum := 1$, $\texttt{CON} = 20\sqrt{q} + \frac{16nq^2}{2^n}$

When $\mathcal{A}$ asks query $\mathsf{LTGen}^{\mathsf{RO},\mathsf{P}}(M)$:
  $UX := \mathsf{RO}(M)$
  $V := \mathsf{P}_1(U)$, $Y := V \oplus X$, $T := \mathsf{P}_2(Y)$
  Add $M$ to TGened
  Return $(T, U, V, X, Y)$

When $\mathcal{A}$ asks query $\mathsf{LVrfy}^{\mathsf{RO},\mathsf{P}}(M, T)$:
  $UX := \mathsf{RO}(M)$
  $Y' := \mathsf{P}_2^{-1}(T)$, $V' := X \oplus Y'$, $U' := \mathsf{P}_1^{-1}(V')$
  Return $(b, U, X, U', V', Y')$, where $b = 1$ if $U = U'$, and $b = 0$ otherwise

When $\mathcal{A}$, $\mathsf{LTGen}^{\mathsf{RO},\mathsf{P}}(M)$, or $\mathsf{LVrfy}^{\mathsf{RO},\mathsf{P}}(M, T)$ asks query $\mathsf{RO}(M)$:
  If $\nexists U, X : (M, UX, \star) \in$ ROSet then
    (1) $UX \xleftarrow{\$} \{0,1\}^{2n}$
    (2) Add $(M, UX, qnum)$ to ROSet
    (3) $qnum := qnum + 1$
    (4) (B-1) If $\exists (M', U'X', \star) \in$ ROSet : $M \neq M'$ and $UX = U'X'$ then abort
    (5) (B-2) If $\exists ((M', U'X', \star), (M'', U''X'', \star)) \in (\text{ROSet})^2 : M \neq M' \neq M''$, and: $U = U' = U''$,
        or $X = X' = X''$ then abort
    (6) (B-3) If $\gamma_1 \geq \sqrt{q}$ then abort     // see Eq. (6) for the definition of $\gamma_1$
    (7) (B-4) If $\gamma_2 \geq \sqrt{q}$ then abort     // see Eq. (7) for the definition of $\gamma_2$
    (8) (B-5) If $\alpha \geq 2\sqrt{q}$ then abort     // see Eq. (4) for the definition of $\alpha$
    (9) (B-6) If $\beta \geq \texttt{CON}$ then abort     // see Eq. (5) for the definition of $\beta$
  EndIf
  Find $U, X$ such that $(M, UX, \star) \in$ ROSet and return $UX$

Private procedure $\mathsf{P}_1(U)$
  If $\nexists V : (U, V, \star, \star) \in$ PSet$_1$ then
    (1) $V \xleftarrow{\$} \{0,1\}^n$ such that
        $\nexists U' : (U', V, \star, \star) \in$ PSet$_1$
    (2) Add $(U, V, \rightarrow, qnum)$ to PSet$_1$
    (3) $qnum := qnum + 1$
    (4) (B-5) If $\alpha \geq 2\sqrt{q}$ then abort
  EndIf
  Find $V$ such that $(U, V, \star, \star) \in$ PSet$_1$ and
return $V$

Private procedure $\mathsf{P}_1^{-1}(V)$
  If $\nexists U : (U, V, \star, \star) \in$ PSet$_1$ then
    (1) $U \xleftarrow{\$} \{0,1\}^n$ such that
        $\nexists V' : (U, V', \star, \star) \in$ PSet$_1$
    (2) Add $(U, V, \leftarrow, qnum)$ to PSet$_1$
    (3) $qnum := qnum + 1$
    (4) (B-5) If $\alpha \geq 2\sqrt{q}$ then abort
  EndIf
  Find $U$ such that $(U, V, \star, \star) \in$ PSet$_1$ and
return $U$

Private procedure $\mathsf{P}_2(Y)$
  If $\nexists T : (Y, T, \star, \star) \in$ PSet$_2$ then
    (1) $T \xleftarrow{\$} \{0,1\}^n$ such that
        $\nexists Y' : (Y', T, \star, \star) \in$ PSet$_2$
    (2) Add $(Y, T, \rightarrow, qnum)$ to PSet$_2$
    (3) $qnum := qnum + 1$
    (4) (B-6) If $\beta \geq \texttt{CON}$ then abort
  EndIf
  Find $T$ such that $(Y, T, \star, \star) \in$ PSet$_2$ and
return $T$

Private procedure $\mathsf{P}_2^{-1}(T)$
  If $\nexists Y : (Y, T, \star, \star) \in$ PSet$_2$ then
    (1) $Y \xleftarrow{\$} \{0,1\}^n$ such that
        $\nexists T' : (Y, T', \star, \star) \in$ PSet$_2$
    (2) Add $(Y, T, \leftarrow, qnum)$ to PSet$_2$
    (3) $qnum := qnum + 1$
    (4) (B-6) If $\beta \geq \texttt{CON}$ then abort
  EndIf
  Find $Y$ such that $(Y, T, \star, \star) \in$ PSet$_2$ and
return $Y$
```

Figure 3: Security game capturing the interaction between $\mathcal{A}$ and the idealized tag generation and verification oracles of LRWHM.

Similarly for (B-4) by symmetry:

$$\Pr[(\text{B-4})] \leq \frac{q^{3/2}}{2^n}. \tag{10}$$

For (B-5), we divide the colliding query-pairs into two disjoint subsets, i.e.,

$$\begin{aligned}\mathcal{S}_1 :=& \Big\{((M, UX, n_1), (U', V', \rightarrow, n_2)) \in \text{ROSet} \times \text{PSet}_1 : U = U', M \notin \text{TGened}, \text{ and } n_2 > n_1\Big\},\\ \mathcal{S}_2 :=& \Big\{((M, UX, n_1), (U', V', \rightarrow, n_2)) \in \text{ROSet} \times \text{PSet}_1 : U = U', M \notin \text{TGened}, \text{ and } n_1 > n_2\Big\}\\ &\cup \Big\{((M, UX, n_1), (U', V', \leftarrow, n_2)) \in \text{ROSet} \times \text{PSet}_1 : U = U' \text{ and } M \notin \text{TGened}\Big\}.\end{aligned}$$

We bound $|\mathcal{S}_1|$ first. According to the code and the conditions, $(U', V', \rightarrow, n_2)$ is necessarily created during processing a query $\mathsf{LTGen}^{\mathsf{RO},\mathsf{P}}(M')$, for which $M' \neq M$ and a corresponding query $(M', UX', \star)$ exists. Conditioned on $\neg$(B-3), the number of RO queries $(M, UX, \star)$ such that $\exists(M', UX', \star)$ is at most $\sqrt{q}$. By this, $|\mathcal{S}_1| \leq \sqrt{q}$.

We then bound $|\mathcal{S}_2|$. For any such pair $((M, UX, n_1), (U', V', dir, n_2))$, we distinguish two cases:

- Case 1: $n_1 > n_2$. Then right before (M, UX, n_1) is created, U is uniform in $\{0,1\}^n$, and thus $\Pr[U = U'] = \frac{1}{2^n}$;
- Case 2: $n_2 > n_1$ and $dir = \leftarrow$. Then right before $(U', V', \leftarrow, n_2)$ is created, U' is uniform in at least $2^n - q$ possibilities, and thus $\Pr[U' = U] \leq \frac{1}{2^n - q} \leq \frac{2}{2^n}$.

Since the number of such pairs $((M, UX, n_1), (U', V', dir, n_2))$ is at most q^2, we can use Markov inequality to obtain

$$\Pr\left[|\mathcal{S}_2| \geq \sqrt{q}\right] \leq \frac{2q^2}{2^n\sqrt{q}} \leq \frac{2q^{3/2}}{2^n}.$$

With $\alpha = |\mathcal{S}_1| + |\mathcal{S}_2|$, we obtain

$$\Pr[(\text{B-5}) \mid \neg(\text{B-3})] \leq \frac{2q^{3/2}}{2^n}. \tag{11}$$

Finally, conditioned on $\neg$(B-5), we analyze (B-6). To derive a bound for β, we view the execution as a random process of "creating" V values. Each time a new record $(M, UX, \star)$ or $(Y, T, \star, \star)$ is added to the sets, a set of V values are "created". Now consider a certain value V. Its frequency μ_V could be increased due to the following three actions:

- a record $(M, UX, \star)$ is added to ROSet;
- a record $(Y, T, \leftarrow, \star)$ is added to PSet_2;
- a record $(Y, T, \rightarrow, \star)$ is added to PSet_2.

We denote by $\mathcal{S}$ the set of all these actions, and divide it into two subsets $\mathcal{S}_3$ and $\mathcal{S}_4$. Intuitively, any record $(Y, T, \rightarrow, n_2) \in \text{PSet}_2$ is in $\mathcal{S}_3$, if and only if:

(i) it is created during processing a tag generation query $\mathsf{LTGen}^{\mathsf{RO},\mathsf{P}}(M)$, and

(ii) for the query $\mathsf{LTGen}^{\mathsf{RO},\mathsf{P}}(M)$, for $UX = \mathsf{RO}(M)$, it holds $(U, V, \star, n_1) \in \text{PSet}_1$ before $\mathsf{LTGen}^{\mathsf{RO},\mathsf{P}}(M)$ is made.

Denote by n_M the value of $qnum$ when $\mathsf{LTGen}^{\mathsf{RO},\mathsf{P}}(M)$ is made, then the two conditions imply that $n_2 = n_M$ or $n_2 = n_M + 1$.[9] Therefore, we have the formal definitions:

$$\begin{aligned}\mathcal{S}_3 := \Big\{&(Y, T, \rightarrow, n_2) \in \mathrm{PSet}_2 : n_2 = n_M \text{ or } n_2 = n_M + 1, \text{and that} \\ &\quad\text{for } UX = \mathsf{RO}(M), (U, V, \star, n_1) \in \mathrm{PSet}_1, \text{ it holds } n_1 < n_M.\Big\} \\ \mathcal{S}_4 := \mathcal{S}&\backslash\mathcal{S}_3.\end{aligned}$$

First, we denote by $\mu_V(\mathcal{S}_3)$ the number of increments on μ_V due to $\mathcal{S}_3$ and analyze it. It's not hard to see $|\mathcal{S}_3| \le \alpha$. Assume that right before such a record $(Y, T, \rightarrow, n_2)$ is added to PSet_2, there have been s different values X in the set ROSet, i.e.,

$$\big|\{X \mid \exists (M, U) : (M, UX, \star) \in \mathrm{ROSet}\}\big| = s.$$

For convenience, we write them as $X_1, \ldots, X_s$. Then it can be seen after $(Y, T, \rightarrow, \star)$ is added to PSet_2, s *distinct* V values $X_1 \oplus Y, \ldots, X_s \oplus Y$ are introduced. By this, $\mu_V(\mathcal{S}_3)$ gets at most 1 chance of increasing. Wlog assume $V = X_i \oplus Y$. Conditioned on $\neg$(B-2), the number of RO query-records $(M, UX, \star)$ such that $X = X_i$ is at most 2; conditioned on $\neg$(B-5), we have $|\mathcal{S}_3| \le \alpha \le 2\sqrt{q}$. Therefore,

$$\mu_V(\mathcal{S}_3) \le 2|\mathcal{S}_3| \le 4\sqrt{q}. \tag{12}$$

We then analyze the increments due to $\mathcal{S}_4$—denoted $\mu_V(\mathcal{S}_4)$. Assume that $|\mathcal{S}_4| = w$. In this respect, we consider a sequence of variables $L_i(V)$, $1 \le i \le w$, where $L_i(V) = 1$ if μ_V is increased during the i-th record creating action (as we have seen, the increment may be greater than 1).

We next prove for any sequence $(a_1, \ldots, a_{i-1}) \in \{0,1\}^{i-1}$, when $q \le 2^n/2$, regardless of the concrete record being created, we have

$$\begin{aligned}\Pr[L_i(V) = 1 \mid (a_1, \ldots, a_{i-1})] \;&:=\; \Pr[L_i(V) = 1 \mid (L_1(V), \ldots, L_{i-1}(V)) = (a_1, \ldots, a_{i-1})] \\ &\le\; \frac{2q}{2^n}.\end{aligned}$$

To this end, we assume that conditioned on $(L_1(V), \ldots, L_{i-1}(V)) = (a_1, \ldots, a_{i-1})$, there have been s different values X and t different values T in the sets, i.e., $\big|\{X \mid \exists (M, U) : (M, UX, \star) \in \mathrm{ROSet}\}\big| = s$ and $\big|\{Y \mid \exists T : (Y, T, \star, \star) \in \mathrm{PSet}_2\}\big| = t$. For convenience, we write these values as $X_1, \ldots, X_s$; $Y_1, \ldots, Y_t$. Now consider a relevant record-creating action. There are three cases:

Case 1: $(M, UX, \star)$ **is created.** In this case, X is uniform in $\{0,1\}^n$ and is independent from the values in the history. And it would create t distinct random V values, i.e.,

$$V_1 = X \oplus Y_1, V_2 = X \oplus Y_2, \ldots, V_t = X \oplus Y_t.$$

Therefore,

$$\Pr[L_i(V) = 1 \mid (a_1, \ldots, a_{i-1})] = \Pr[X \in \{V \oplus Y_1, \ldots, V \oplus Y_t\}] = \frac{t}{2^n} \le \frac{q}{2^n} \le \frac{2q}{2^n}.$$

Case 2: $(Y, T, \leftarrow, \star)$ **is created.** In this case, Y is uniform in $\{0,1\}^n \backslash \{Y_1, \ldots, Y_t\}$, and s distinct random V values are created, i.e.,

$$V_1 = X_1 \oplus Y, V_2 = X_2 \oplus Y, \ldots, V_s = X_s \oplus Y.$$

[9] After the query $\mathsf{LTGen}^{\mathsf{RO},\mathsf{P}}(M)$ is made, (a) if $(M, UX, \star) \in \mathrm{ROSet}$, then since we assumed $(U, V, \star, n_1)$ existed before, the game just creates the record $(Y, T, \rightarrow, n_2)$ with $n_2 = n_M$; and (b) if $(M, UX, \star) \notin \mathrm{ROSet}$, then (M, UV, n_M) is created for some UV. Then, since we assumed that $(U, V, \star, n_1)$ existed before, the game creates the record $(Y, T, \rightarrow, n_2)$ with $n_2 = n_M + 1$.

For each $i \in \{1, \ldots, s\}$, when $V \oplus X_i \in \{Y_1, \ldots, Y_t\}$ we have $\Pr[Y \oplus X_i = V] = 0$, otherwise $\Pr[Y \oplus X_i = V] = \frac{1}{2^n - t} \leq \frac{1}{2^n - q}$. Therefore,

$$\Pr[L_i(V) = 1 \mid (a_1, \ldots, a_{i-1})] = \Pr[Y \in \{V \oplus X_1, \ldots, V \oplus X_s\}] \leq \frac{s}{2^n - q} \leq \frac{2q}{2^n}.$$

Case 3: $(Y, T, \rightarrow, \star)$ is created. This record has to be created during processing a query $\mathsf{LTGen}^{\mathsf{RO},\mathsf{P}}(M')$. And right before $(Y, T, \rightarrow, \star) \in \mathrm{PSet}_2$ holds, there exists $((M', U'X', \star), (U', V', \star, \star)) \in \mathrm{ROSet} \times \mathrm{PSet}_1$ such that $Y = V' \oplus X'$. By the definition of $\mathcal{S}_4$, right before $(Y, T, \rightarrow, \star) \in \mathrm{PSet}_2$ holds, it holds $(U', V', \rightarrow, \star) \in \mathrm{PSet}_1$. Then V' is uniform in a set which we denote $\mathcal{V}$ for convenience, which satisfies $|\mathcal{V}| \geq 2^n - q$. This means $Y = V' \oplus X'$ is uniform in the set $X' \oplus \mathcal{V}$. In a similar vein to Case 2, we have $\Pr[L_i(V) = 1 \mid (a_1, \ldots, a_{i-1})] \leq \frac{2q}{2^n}$.

Conditioned on $\neg$(B-2), for any certain value X_i, the number of RO queries $(M, UX, \star)$ such that $X = X_i$ is at most 2. By this and the above, we have

$$\mu_V(\mathcal{S}_4) \leq 2 \sum_{i=1}^{w} L_i(V). \tag{13}$$

Via a Chernoff bound-based argument that follows [CLL+18, PS15], it can be proved

$$\Pr\left[\exists V \in \{0,1\}^n : \sum_{i=1}^{w} L_i(V) \geq 8\sqrt{q} + \frac{8nq^2}{2^n}\right] \leq \frac{1}{2^n}. \tag{14}$$

The proof is deferred to Appendix A for cleanness. Eq. (12) plus (13) and (14) indicate the constant $\mathtt{CON} = 20\sqrt{q} + \frac{16nq^2}{2^n}$ constitutes an upper bound on μ_V for any V, i.e.

$$\begin{aligned}
&\Pr[(\text{B-6}) \mid \neg(\text{B-2}) \wedge \neg(\text{B-5})] \\
=&\Pr\left[\beta \geq 20\sqrt{q} + \frac{16nq^2}{2^n} \mid \neg(\text{B-2}) \wedge \neg(\text{B-5})\right] \\
=&\Pr\left[\exists V \in \{0,1\}^n : \mu_V \geq 20\sqrt{q} + \frac{16nq^2}{2^n} \mid \neg(\text{B-2}) \wedge \neg(\text{B-5})\right] \leq \frac{1}{2^n} \leq \frac{q^{3/2}}{2^n}.
\end{aligned} \tag{15}$$

Gathering Eq. (8), (9), (10), (11), and (15) yields

$$\Pr\left[\mathcal{A}^{\mathsf{RO},\mathsf{LTGen}^{\mathsf{RO},\mathsf{P}},\mathsf{LVrfy}^{\mathsf{RO},\mathsf{P}}} \text{ aborts}\right] \leq \frac{q^{3/2}}{2^n} + 2 \cdot \frac{q^{3/2}}{2^n} + \frac{2q^{3/2}}{2^n} + \frac{q^{3/2}}{2^n} \leq \frac{6q^{3/2}}{2^n}. \tag{16}$$

3.1.3 Unforgeability Unless Abortion

If the action $\mathsf{LTGen}^{\mathsf{RO},\mathsf{P}}(M) \rightarrow T$ happens, then all the subsequent non-trivial verification queries $\mathsf{LVrfy}^{\mathsf{RO},\mathsf{P}}(M, T')$ have to satisfy $T' \neq T$. Since P_1 and P_2 are two permutations, it always holds $\mathsf{LVrfy}^{\mathsf{RO},\mathsf{P}}(M, T') \neq 1$. Therefore, we could concentrate on the verification queries $\mathsf{LVrfy}^{\mathsf{RO},\mathsf{P}}(M, T')$ for which $\mathsf{LTGen}^{\mathsf{RO},\mathsf{P}}(M)$ was *never made*. To this end, we define an event Chain capturing the aforementioned “chains of records”: at any time during the execution, there exist three query-records $(M, UX, n_1) \in \mathrm{ROSet}$, $(U, V, d_1, n_2) \in \mathrm{PSet}_1$, and $(Y, T, d_2, n_3) \in \mathrm{PSet}_2$ such that $Y = V \oplus X$, and $M \notin \mathrm{TGened}$. To complete the analysis, we derive an upper bound on $\Pr[\mathsf{Chain} \mid \neg(\mathcal{A}^{\mathsf{RO},\mathsf{LTGen}^{\mathsf{RO},\mathsf{P}},\mathsf{LVrfy}^{\mathsf{RO},\mathsf{P}}} \text{ aborts})]$. For this, note that the presence of such a chain consists of 5 cases:

- (C-1): $n_1 > n_2, n_3$. Informally, right before a new record $(M, UX, \star)$ is added to ROSet, there already exist $(U, V, \star, \star) \in \mathrm{PSet}_1$ and $(Y, T, \star, \star) \in \mathrm{PSet}_2$ such that $Y = V \oplus X$;

- (C-2): $n_2 > n_1, n_3$, and $d_1 = \rightarrow$. Informally, right before a new query $(U, V, \rightarrow, \star)$ is added to PSet_1, there already exist $(M, UX, \star) \in \text{ROSet}$ and $(Y, T, \star, \star) \in \text{PSet}_2$ such that $V = X \oplus Y$;
- (C-3): $n_2 > n_1, n_3$, and $d_1 = \leftarrow$. Informally, right before a new record $(U, V, \leftarrow, \star)$ is added to PSet_1, there already exist $(M, UX, \star) \in \text{ROSet}$ and $(Y, T, \star, \star) \in \text{PSet}_2$ such that $V = X \oplus Y$;
- (C-4): $n_3 > n_1, n_2$, and $d_2 = \rightarrow$. Informally, right before a new record $(Y, T, \rightarrow, \star)$ is added to PSet_2, there already exist $(M, UX, \star)$ and $(U, V, \star, \star) \in \text{PSet}_1$ such that $Y = V \oplus X$, and $M \notin \text{TGened}$;
- (C-5): $n_3 > n_1, n_2$, and $d_2 = \leftarrow$. Informally, right before a new record $(Y, T, \leftarrow, \star)$ is added to PSet_2, there already exist $(M, UX, \star) \in \text{ROSet}$ and $(U, V, \star, \star) \in \text{PSet}_1$ such that $Y = V \oplus X$.

In addition, after the involved action is completed, it remains $M \notin \text{TGened}$. Below we analyze them in turn.

For (C-1): for each such record $(M, UX, \star)$, right before it's added to ROSet, both U and X are uniform. There are at most q^2 "targets" $((U', V', \star, \star), (Y', T', \star, \star))$. Therefore,

$$\Pr[\text{(C-1)}] \le \frac{|\text{ROSet}| \cdot q^2}{2^{2n}} \le \frac{q^3}{2^{2n}} \le \frac{q^{3/2}}{2^n}.$$

For (C-2): from the code it's easy to see $(U, V, \rightarrow, \star)$ is created during a $\mathsf{LTGen}^{\mathsf{RO},\mathsf{P}}(M')$ query. It has to be

- $M' \neq M$, otherwise the resulted record chain does not satisfy $M \notin \text{TGened}$, and
- for $U'X' = \mathsf{RO}(M')$ it holds $U' = U$.

Conditioned on $\neg$(B-3), the number of choices for the query (M, UX) that has a corresponding M' satisfying such requirements is $\le \sqrt{q}$. For each such (M, UX), right before $(U, V, \rightarrow, \star)$ is added, V is uniform in $\ge 2^n - q$ values; and there are $\le q$ queries $(Y, T, \star, \star)$. Therefore, $\Pr[U = X \oplus Y \text{ for } (Y, T, \star, \star) \in \text{PSet}_2] \le \frac{q}{2^n - q}$, and

$$\Pr[\text{(C-2)} \mid \neg\text{(B-3)}] \le \frac{q^{3/2}}{2^n - q} \le \frac{2q^{3/2}}{2^n}.$$

For (C-3): for each query $\mathsf{P}_1^{-1}(V)$, the number of pairs $((M, UX, \star), (Y, T, \star, \star))$ such that $V = X \oplus Y$ is at most β. This means the number of "target" values U does not exceed β either. For each $\mathsf{P}_1^{-1}(V) \rightarrow U'$, U' is uniform in $\ge 2^n - q$ values; taking a union bound over the $\le q$ queries to P_1^{-1} yields $\Pr[\text{(C-3)}] \le \frac{\beta q}{2^n - q}$. Therefore,

$$\Pr[\text{(C-3)} \mid \neg\text{(B-6)}] \le \frac{q \cdot \mathtt{CON}}{2^n - q} \le \frac{40q^{3/2}}{2^n} + \frac{32nq^3}{2^{2n}}.$$

For (C-4): From the code, $(Y, T, \rightarrow, \star)$ must be created during processing a $\mathsf{LTGen}^{\mathsf{RO},\mathsf{P}}(M')$ query. It has to be $M' \neq M$ otherwise $M \in \text{TGened}$. More importantly, right before $(Y, T, \rightarrow, \star)$ is created, there exist $(M, UX, \star), (M', U'X', \star), (U, V, \star, \star), (U', V', \star, \star)$ such that $X \oplus V = X' \oplus V'$ and $M \notin \text{TGened}$. Moreover, it has to be $U \neq U'$: otherwise $U = U' \Rightarrow X \neq X'$ by $\neg$(B-1) and $X \oplus V = X' \oplus V$ is not possible.

We switch to bound the probability that such four queries appear. It can be seen this relies on the occurrence of the following five events:

- (C-41): the latest query is an RO query. Formally, during processing $\mathsf{RO}(M)$, right after the answer UX is sampled, there already exist 3 queries $(M', U'X', \star), (U, V, \star, \star)$, and $(U', V', \star, \star)$. There are $\le q$ choices for $\mathsf{RO}(M)$, $\le q$ choices for $(U, V, \star, \star)$, and $\le q$ choices for the pairs $((M', U'X', \star), (U', V', \star, \star))$. Therefore, $\Pr[\text{(C-41)}] \le \frac{q^3}{2^{2n}}$;

- (C-42): before creating $(U', V', \to, \star)$, there exist $(M, UX, \star), (U, V, \star, \star), (M', U'X', \star)$ such that $X \oplus V = X' \oplus V'$. Denote by q_1 the number of such forward queries to P_1. Since $M \notin$ TGened, we have $\leq 2\sqrt{q}$ choices for the pair $((M, UX, \star), (U, V, \star, \star))$ conditioned on $\neg$(B-5); and conditioned on $\neg$(B-2), we have ≤ 2 choices for $(M', U'X', \star)$. And V' is uniform in $\geq 2^n - q$ possibilities. Therefore, $\Pr[\text{(C-42)}] \leq \frac{4q_1\sqrt{q}}{2^n-q}$;
- (C-43): before creating $(U, V, \to, \star)$, there exist $(M, UX, \star), (M', U'X', \star), (U', V', \star, \star)$ such that $X \oplus V = X' \oplus V'$. It can be seen $M' \notin$ TGened before creating $(U, V, \to, \star)$, otherwise $(Y, T, \star, \star) \in$ PSet before creating $(U, V, \to, \star)$, and thus the action of creating $(Y, T, \to, \star)$ cannot happen later. Therefore, in a similar vein to (C-42), we have $\Pr[\text{(C-43)}] \leq \frac{4q_1\sqrt{q}}{2^n-q}$;
- (C-44): before creating $(U'', V', \leftarrow, \star)$, there exist $(M, UX, \star), (U, V, \star, \star), (M', U'X', \star)$ such that $U'' = U'$ and $X \oplus V = X' \oplus V'$. Denote by q_2 the number of such queries to P_1^{-1}. By $M \notin$ TGened and $\neg$(B-5), we have $\leq 2\sqrt{q}$ choices for the pair $((M, UX, \star), (U, V, \star, \star))$. For each combination of them, the involved value $X' = X \oplus V \oplus V'$ has been fixed; by $\neg$(B-2), the number of choices for $(M', U'X', \star)$ is ≤ 2. And U'' is uniform in $\geq 2^n - q$ possibilities. Therefore, $\Pr[\text{(C-44)}] \leq \frac{4q_2\sqrt{q}}{2^n-q}$;
- (C-45): before creating $(U'', V, \leftarrow, \star)$, there exist $(M, UX, \star), (M', U'X', \star), (U', V', \star, \star)$ such that $U'' = U$ and $X \oplus V = X' \oplus V'$. Similarly to (C-43), $M' \notin$ TGened before creating $(U'', V, \leftarrow, \star)$. Thus the analysis resembles (C-44), yielding $\Pr[\text{(C-45)}] \leq \frac{4q_2\sqrt{q}}{2^n-q}$.

Note that $q_1 + q_2 \leq q$. Therefore, when $q \leq 2^n/2$,

$$\begin{aligned}\Pr[\text{(C-4)} \mid \neg\text{(B-2)} \wedge \neg\text{(B-5)}] &\leq \sum_{i=1}^{5} \Pr[\text{(C-4}i\text{)} \mid \neg\text{(B-2)} \wedge \neg\text{(B-5)}] \\ &\leq \frac{q^3}{2^{2n}} + 2 \cdot \frac{4q_1\sqrt{q}}{2^n - q} + 2 \cdot \frac{4q_2\sqrt{q}}{2^n - q} \leq \frac{17q^{3/2}}{2^n}.\end{aligned}$$

For (C-5): for a query $\mathsf{P}_2^{-1}(T) \to Y$, if $Y = V \oplus X$ for a pair $((M, UX, \star), (U, V, \star, \star))$, then $M \notin$ TGened before this action: otherwise, there already existed $(Y', T', \star, \star) \in \text{PSet}_2$ that corresponds to the computation of $\mathsf{LTGen}^{\mathsf{RO},\mathsf{P}}(M) \to T'$, so that either $T = T'$ and $(Y, T, \leftarrow, \star) \in \text{PSet}_2$ cannot be newly created, or $T \neq T'$ and further $Y \neq Y' = V \oplus X$, a contradiction. By this, conditioned on $\neg$(B-5), the number of such "target" pairs $((M, UX, \star), (U, V, \star, \star))$ with $M \notin$ TGened is $\alpha < 2\sqrt{q}$, and thus

$$\Pr[\text{(C-5)} \mid \neg\text{(B-5)}] \leq q \cdot \frac{2\sqrt{q}}{2^n - q} \leq \frac{4q^{3/2}}{2^n}.$$

By the above,

$$\begin{aligned}&\Pr[\mathsf{Chain} \mid \neg(\mathcal{A}^{\mathsf{RO},\mathsf{LTGen}^{\mathsf{RO},\mathsf{P}},\mathsf{LVrfy}^{\mathsf{RO},\mathsf{P}}} \text{ aborts})] \\ \leq\ &\Pr[\text{(C-1)} \vee \text{(C-2)} \vee \text{(C-3)} \vee \text{(C-4)} \vee \text{(C-5)} \mid \neg(\mathcal{A}^{\mathsf{RO},\mathsf{LTGen}^{\mathsf{RO},\mathsf{P}},\mathsf{LVrfy}^{\mathsf{RO},\mathsf{P}}} \text{ aborts})] \\ \leq\ &\frac{q^{3/2}}{2^n} + \frac{2q^{3/2}}{2^n} + \frac{40q^{3/2}}{2^n} + \frac{32nq^3}{2^{2n}} + \frac{17q^{3/2}}{2^n} + \frac{4q^{3/2}}{2^n} \\ \leq\ &\frac{32nq^3}{2^{2n}} + \frac{64q^{3/2}}{2^n}.\end{aligned}$$

This further plus Eq. (16) yield

$$\begin{aligned}\mathbf{Adv}^{\mathsf{MAL2}}_{\mathsf{LRWHM}^{\mathsf{RO},\mathsf{P}}}(\mathcal{A}) &\leq \underbrace{\Pr[\mathcal{A}^{\mathsf{RO},\mathsf{LTGen}^{\mathsf{RO},\mathsf{P}},\mathsf{LVrfy}^{\mathsf{RO},\mathsf{P}}} \text{ aborts}]}_{\leq \frac{6q^{3/2}}{2^n}} + \Pr[\mathsf{Chain} \mid \neg(\mathcal{A}^{\mathsf{RO},\mathsf{LTGen}^{\mathsf{RO},\mathsf{P}},\mathsf{LVrfy}^{\mathsf{RO},\mathsf{P}}} \text{ aborts})] \\ &\leq \frac{32nq^3}{2^{2n}} + \frac{70q^{3/2}}{2^n}.\end{aligned}$$

```
$\mathsf{TGen}_K^{\mathsf{H},\mathsf{E}}(M)$: proceeds in two steps:
    1. Computes $U\|X = \mathsf{H}(M)$, $V = \mathsf{E}_K(U)$, and $T = \mathsf{E}_V(X)$;
    2. Return $T$ as the tag. $\underline{\mathsf{L}_{\mathsf{TGen}}(K, M) := (U, X, T).}$

$\mathsf{Vrfy}_K^{\mathsf{H},\mathsf{E}}(M, T)$: proceeds in three steps:
    1. Forward Computation: computes $U\|X = \mathsf{H}(M)$ and $V = \mathsf{E}_K(U)$;
    2. Backward Computation: computes $X' = \mathsf{E}_V^{-1}(T)$;
    3. If $X = X'$, then return 1, else return 0. $\underline{\mathsf{L}_{\mathsf{Vrfy}}(K, M, T) := (U, X, X').}$
```

Figure 4: The description of $\mathsf{RHM}^{\mathsf{H},\mathsf{E}}$ MA mode.

This plus the gap $2\mathbf{Adv}_{\mathsf{E}}^{\mathsf{SPRP}}(\mathcal{A}')$ result in Eq. (2). □

Under the "unbounded leakage" assumption, the provable bound $2^{2n/3}/n$ is tight, as we will justify in Appendix B. But we are not aware of any attack with low data and feasible time complexity, i.e., any attack cheaper than the naïve side-channel key recovery. Deeper characterization of the concrete side-channel security is left for future work.

3.2 Mode RHM and Its Security

Formally, the mode $\mathsf{RHM}^{\mathsf{H},\mathsf{E}}$ along with the leakages is defined in Fig. 4. In the ideal cipher model, the $\mathsf{MAL2}$ security of RHM is up to $2^n/n$ queries.

Theorem 2. *Let $\mathsf{IC} : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}^n$ be an ideal cipher and $\mathsf{RO} : \{0,1\}^* \to \{0,1\}^{2n}$ be a random oracle, and assume $q \le 2^n/2$. Then with the leakages $\mathsf{L}_{\mathsf{TGen}}$ and $\mathsf{L}_{\mathsf{Vrfy}}$ (Fig. 4), for any (q,t)-adversary $\mathcal{A}$ against the $\mathsf{MAL2}$ security of $\mathsf{RHM}^{\mathsf{RO},\mathsf{IC}}$, it holds*

$$\mathbf{Adv}_{\mathsf{RHM}^{\mathsf{RO},\mathsf{IC}}}^{\mathsf{MAL2}}(\mathcal{A}) \le \frac{3q^2}{2^{2n+1}} + \frac{4nq}{2^n}. \tag{17}$$

Proof. Similarly to Theorem 1, the optimal security of RHM cannot be obtained via a modular approach, as the TBC $\widetilde{\mathsf{E}}(tw, X) = \mathsf{E}_{\mathsf{E}_K(tw)}(X)$ is not BBB secure [Min09]. Therefore, we divide our analysis into an overview and two steps as below.

3.2.1 Proof Overview

The proof flow is very similar to Theorem 1. Concretely, we first idealize the scheme $\mathsf{RHM}_K^{\mathsf{RO},\mathsf{IC}}$ as $\mathsf{RHM}^{\mathsf{RO},\mathsf{P},\mathsf{IC}}$, in which the first call to IC_K is replaced by a random permutation P that is never queried by the adversary $\mathcal{A}$. The difference between $\mathbf{Adv}_{\mathsf{RHM}^{\mathsf{RO},\mathsf{IC}}}^{\mathsf{MAL2}}(\mathcal{A})$ and $\mathbf{Adv}_{\mathsf{RHM}^{\mathsf{RO},\mathsf{P},\mathsf{IC}}}^{\mathsf{MAL2}}(\mathcal{A})$ is reduced to the PRP security of the ideal cipher IC. Unless the adversary hits the key K in its q queries to IC, the two systems $(\mathsf{IC}_K, \mathsf{IC})$ and $(\mathsf{P}, \mathsf{IC})$ are indistinguishable. For each adversarial query to IC, the probability of such a "hit" is $1/2^n$. Summing over the q adversarial queries, we reach

$$\left|\mathbf{Adv}_{\mathsf{RHM}^{\mathsf{RO},\mathsf{IC}}}^{\mathsf{MAL2}}(\mathcal{A}) - \mathbf{Adv}_{\mathsf{RHM}^{\mathsf{RO},\mathsf{P},\mathsf{IC}}}^{\mathsf{MAL2}}(\mathcal{A})\right| \le \frac{q}{2^n}. \tag{18}$$

We then focus on bounding $\mathbf{Adv}_{\mathsf{RHM}^{\mathsf{RO},\mathsf{P},\mathsf{IC}}}^{\mathsf{MAL2}}(\mathcal{A})$. We also describe the security game using pseudocode in Fig. 5. The sets, the query-records, the auxiliary variables (dir and $qnum$), and the abort mechanism are all similar to Fig. 3. At any time during the interaction, given the internal sets, we define three auxiliary sets

$$RO[U] := \{(M, X) : (M, UX, \star) \in \text{ROSet}\}, \quad RO[X] := \{(M, U) : (M, UX, \star) \in \text{ROSet}\},$$
$$IC[X] := \{(V, T) : (V, X, T, \star, \star) \in \text{ICSet}\}.$$

The remaining two steps also resemble subsection 3.1: first, we bound the probability that $\mathcal{A}^{\mathsf{RO},\mathsf{LTGen}^{\mathsf{RO},\mathsf{P},\mathsf{IC}},\mathsf{LVrfy}^{\mathsf{RO},\mathsf{P},\mathsf{IC}}}$ aborts; second, we show that LVrfy always returns 0 if $\mathcal{A}^{\mathsf{RO},\mathsf{LTGen}^{\mathsf{RO},\mathsf{P},\mathsf{IC}},\mathsf{LVrfy}^{\mathsf{RO},\mathsf{P},\mathsf{IC}}}$ doesn't abort.

3.2.2 Probability of Abortion

Consider (C-1) first. Note that the set PSet defines a one-to-one correspondence. This means for each $(V, X, T, \star, \star) \in$ ICSet, the number of U such that $(U, V, \star) \in$ PSet is at most 1. Therefore, the number of "target" pairs $((U, V, \star), (V, X, T, \star, \star))$ is $\leq |\text{ICSet}| \leq q$, and

$$\Pr[\text{(C-1)}] \leq |\text{ROSet}| \cdot \frac{|\text{ICSet}|}{2^{2n}} \leq \frac{q^2}{2^{2n}}.$$

(C-2) essentially states that RO-collisions occur. So $\Pr[\text{(C-2)}] \leq \frac{q^2}{2^{2n+1}}$.

(C-3) essentially states that there exist n queries $(M_1, U_1X_1, \star), \ldots, (M_n, U_nX_n, \star)$ such that $U_1 = \ldots = U_n$. The number of choices for such n queries is $\binom{q}{n}$; for any of them, the probability to have $U_1 = \ldots = U_n$ is $1/(2^n)^{n-1}$. So

$$\Pr[\text{(C-3)}] \leq \binom{q}{n} \cdot \frac{1}{(2^n)^{n-1}} \leq \frac{1}{n!} \cdot \left(\frac{q}{2^{n-1}}\right)^n \leq \frac{2q}{2^n}.$$

Similarly for (C-4) by symmetry: $\Pr[\text{(C-4)}] \leq \frac{2q}{2^n}$.

For (C-5), consider such a query $\mathsf{IC}(V, X)$. We distinguish two cases. In the first case, $\mathsf{IC}(V, X)$ is made during processing a tag query $\mathsf{LTGen}^{\mathsf{RO},\mathsf{P},\mathsf{IC}}(M)$. Let $UX = \mathsf{RO}(M)$. Then conditioned on $\neg$(C-2), for any $M' \neq M$ and $U'X' = \mathsf{RO}(M')$ such that $X' = X$, it necessarily holds $U \neq U'$. Since PSet defines a bijection, this means $V' = \mathsf{P}(U') \neq V$, i.e., in this case, (C-5) would *not* be triggered.

In the second case, $\mathsf{IC}(V, X)$ is made by the adversary. Then, conditioned on $\neg$(C-4), the number of (M, U) such that $(M, UX, \star) \in$ ROSet is at most $n-1$. For each such U, if there exists V such that $(U, V, \star) \in$ PSet, then since V is never leaked to $\mathcal{A}$, conditioned on the transcript of queries and answers (including leakages) obtained by $\mathcal{A}$, V remains uniform in at least $2^n - q$ possibilities (since it does not equal V' for any $(V', X', T', \star, \star) \in$ ICSet that is known to $\mathcal{A}$). Therefore, for each query $\mathsf{IC}(V, X)$, the probability of abortion is at most $\frac{n-1}{2^n - q}$. Denote by q_1 the number of such forward queries to IC, then

$$\Pr[\text{(C-5)} \mid \neg\text{(C-2)} \wedge \neg\text{(C-4)}] \leq \frac{(n-1)q_1}{2^n - q} \leq \frac{2(n-1)q_1}{2^n}.$$

For (C-6), consider such a query $\mathsf{IC}^{-1}(V, T)$. As PSet defines a bijection, the number of corresponding U is at most 1. Then, conditioned on $\neg$(C-3), the number of (M, X') such that $(M, UX', \star) \in$ ROSet is at most $n - 1$. For each such $(M, UX', \star)$, since the newly sampled $X = \mathsf{IC}^{-1}(V, T)$ is uniform in $\geq 2^n - q$ possibilities, $\Pr[X = X'] \leq \frac{1}{2^n - q}$. By these, denote by q_2 the number of such backward queries to IC^{-1}, then

$$\Pr[\text{(C-6)} \mid \neg\text{(C-3)}] \leq q_2 \cdot \frac{n-1}{2^n - q} \leq \frac{2(n-1)q_2}{2^n}.$$

Using $q_1 + q_2 \leq |\text{ICSet}| \leq q$ we obtain

$$\Pr[\text{(C-5)} \vee \text{(C-6)} \mid \neg\text{(C-2)} \wedge \neg\text{(C-3)} \wedge \neg\text{(C-4)}] \leq \frac{2(n-1)q}{2^n}.$$

Finally consider (C-7). Consider the i-th such query $\mathsf{P}(U_i)$ that samples V_i. For clearness, write

$$RO[U_i] = \big\{(M_{i,1}, U_i \| X_{i,1}, \star), \ldots, (M_{i,\alpha_i}, U_i \| X_{i,\alpha_i}, \star)\big\}.$$

```
Initialization:
  $\text{ROSet} := \emptyset$, $\text{TGened} := \emptyset$, $\text{PSet} := \emptyset$, $\text{ICSet} := \emptyset$, $qnum := 1$

When $\mathcal{A}$ asks query $\mathsf{LTGen}^{\mathsf{RO},\mathsf{P},\mathsf{IC}}(M)$:
  Add $M$ to TGened
  $UX := \mathsf{RO}(M)$
  $V := \mathsf{P}(U)$, $T := \mathsf{IC}(V, X)$
  Return $(T, U, X)$

When $\mathcal{A}$ asks query $\mathsf{LVrfy}^{\mathsf{RO},\mathsf{P},\mathsf{IC}}(M, T)$:
  $UX := \mathsf{RO}(M)$
  $V := \mathsf{P}(U)$, $X' := \mathsf{IC}^{-1}(V, T)$
  Return $(b, U, X, X')$, where $b = 1$ if $X = X'$, and $b = 0$ otherwise

When $\mathcal{A}$, $\mathsf{LTGen}^{\mathsf{RO},\mathsf{P},\mathsf{IC}}(M)$, or $\mathsf{LVrfy}^{\mathsf{RO},\mathsf{P},\mathsf{IC}}(M, T)$ asks query $\mathsf{RO}(M)$:
  If $\nexists U, X : (M, UX, \star) \in \text{ROSet}$ then
    (1) $UX \xleftarrow{\$} \{0,1\}^{2n}$
    (2) (C-1) If $\exists V, T : (U, V, \star) \in \text{PSet}$ and $(V, X, T, \star, \star) \in \text{ICSet}$ then abort
    (3) (C-2) If $\exists M' : (M', UX, \star) \in \text{ROSet}$ then abort
    (4) Add $(M, UX, qnum)$ to ROSet
    (5) $qnum := qnum + 1$
    (6) (C-3) If $\exists U : \big|RO[U]\big| \geq n$ then abort
    (7) (C-4) If $\exists X : \big|RO[X]\big| \geq n$ then abort
  EndIf
  Find $U, X$ such that $(M, UX, \star) \in \text{ROSet}$ and return $UX$

When $\mathcal{A}$, $\mathsf{LTGen}^{\mathsf{RO},\mathsf{P},\mathsf{IC}}(M)$, or $\mathsf{LVrfy}^{\mathsf{RO},\mathsf{P},\mathsf{IC}}(M, T)$ asks query $\mathsf{IC}(V, X)$:
  If $\nexists T : (V, X, T, \star, \star) \in \text{ICSet}$ then
    (1) $T \xleftarrow{\$} \{0,1\}^n$ such that $\nexists X' : (V, X', T, \star, \star) \in \text{ICSet}$
    (2) (C-5) If $\exists M, U : (U, V, \star) \in \text{PSet}$, $(M, UX, \star) \in \text{ROSet}$, and $M \notin \text{TGened}$ then abort
    (3) Add $(V, X, T, \rightarrow, qnum)$ to ICSet
    (4) $qnum := qnum + 1$
  EndIf
  Find $T$ such that $(V, X, T, \star, \star) \in \text{ICSet}$ and return $T$

When $\mathcal{A}$, $\mathsf{LTGen}^{\mathsf{RO},\mathsf{P},\mathsf{IC}}(M)$, or $\mathsf{LVrfy}^{\mathsf{RO},\mathsf{P},\mathsf{IC}}(M, T)$ asks query $\mathsf{IC}^{-1}(V, T)$:
  If $\nexists X : (V, X, T, \star, \star) \in \text{ICSet}$ then
    (1) $X \xleftarrow{\$} \{0,1\}^n$ such that $\nexists T' : (V, X, T', \star, \star) \in \text{ICSet}$
    (2) (C-6) If $\exists M, U : (U, V, \star) \in \text{PSet}$ and $(M, UX, \star) \in \text{ROSet}$ then abort
    (3) Add $(V, X, T, \leftarrow, qnum)$ to ICSet
    (4) $qnum := qnum + 1$
  EndIf
  Find $X$ such that $(V, X, T, \star, \star) \in \text{ICSet}$ and return $X$

Private procedure $\mathsf{P}(U)$
  If $\nexists V : (U, V, \star) \in \text{PSet}$ then
    (1) $V \xleftarrow{\$} \{0,1\}^n$ such that $\nexists U' : (U', V, \star) \in \text{PSet}$
    (2) (C-7) If $\exists M, X, T : (M, UX, \star) \in \text{ROSet}$ and $(V, X, T, \star, \star) \in \text{ICSet}$ then abort
    (3) Add $(U, V, qnum)$ to PSet
    (4) $qnum := qnum + 1$
  EndIf
  Find $V$ such that $(U, V, \star) \in \text{PSet}$ and return $V$
```

Figure 5: Security game capturing the interaction between $\mathcal{A}$ and the idealized tag generation and verification oracles of RHM.

By $\neg$(C-3), $\alpha_i \le n-1$; then it can be seen

$$\begin{aligned}\Pr[(\text{C-7}) \mid \neg(\text{C-3})] &\le \sum_{i=1}^{q} \Pr\left[\exists T \text{ and } j \in \{1,\ldots,\alpha_i\} : (V_i,T) \in IC[X_{i,j}]\right] \\ &\le \frac{\sum_{i=1}^{q}\sum_{j=1}^{\alpha_i}\left|IC[X_{i,j}]\right|}{2^n - q}\end{aligned}$$

Conditioned on $\neg$(C-2) and $\neg$(C-4), in the summation $\sum_{i=1}^{q}\sum_{j=1}^{\alpha_i}\left|IC[X_{i,j}]\right|$, each specific X appears at most $n-1$ times. Therefore, there exists a set $\mathcal{X}$ of n-bit values such that

$$\sum_{i=1}^{q}\sum_{j=1}^{\alpha_i}\left|IC[X_{i,j}]\right| \le (n-1)\cdot \sum_{X\in\mathcal{X}}\left|IC[X_{i,j}]\right|.$$

On the other hand, for any such set $\mathcal{X}$ we have $\sum_{X\in\mathcal{X}}\left|IC[X_{i,j}]\right| \le \left|\text{ICSet}\right| \le q$. Therefore,

$$\Pr[(\text{C-7}) \mid \neg(\text{C-2}) \wedge \neg(\text{C-3}) \wedge \neg(\text{C-4})] \le \frac{(n-1)q}{2^n - q} \le \frac{2(n-1)q}{2^n}.$$

By all the above, when $q \le 2^n/2$,

$$\begin{aligned}\Pr[\mathcal{A}^{\mathsf{RO},\mathsf{LTGen}^{\mathsf{RO,P,IC}},\mathsf{LVrfy}^{\mathsf{RO,P,IC}}} \text{ aborts}] &\le \frac{q^2}{2^{2n}} + \frac{q^2}{2^{2n+1}} + 2\cdot\frac{2q}{2^n} + 2\cdot\frac{2(n-1)q}{2^n} \\ &\le \frac{3q^2}{2^{2n+1}} + \frac{(4n-1)q}{2^n}.\end{aligned}$$

3.2.3 Unforgeability Unless Abortion

Consider the Chain event: at any time during the interaction, there exists three query-records (M, UX, n_1), (U, V, n_2), and (V, X, T, dir, n_3) such that $M \notin$ TGened.

- If $n_1 > n_2, n_3$, i.e., (M, UX, n_1) is created latest, then it contradicts $\neg$(C-1);
- If $n_2 > n_1, n_3$, then it contradicts $\neg$(C-7);
- If $n_3 > n_1, n_2$ and $dir = \leftarrow$, then it contradicts $\neg$(C-6);
- Else, $n_3 > n_1, n_2$ and $dir = \rightarrow$, then it contradicts $\neg$(C-5).

By the above, the Chain event is impossible in non-aborting executions, and thus

$$\mathbf{Adv}^{\mathsf{MAL2}}_{\mathsf{RHM}^{\mathsf{RO,P,IC}}}(\mathcal{A}) \le \Pr[\mathcal{A}^{\mathsf{RO},\mathsf{LTGen}^{\mathsf{RO,P,IC}},\mathsf{LVrfy}^{\mathsf{RO,P,IC}}} \text{ aborts}] \le \frac{3q^2}{2^{2n+1}} + \frac{(4n-1)q}{2^n}. \quad (19)$$

This plus the term $q/2^n$ in Eq. (18) yield (17). □

4 Performance Evaluations

In this section we report our implementation results. The blockcipher in our modes is naturally instantiated with AES128. More specifically, we follow [GR17] (concretely, the "KHL method") and implement the masked AES with various orders in C code.[10] On the other hand, the hash functions are instantiated with the SHA3 variant built upon 16-round KECCAK-f[400] implementations of [DEM+19] rather than the standard KECCAK-f[1600] to enable a fair comparison to ISAPMACKA. We'll refer to these instances of LRWHM and RHM by AES-SHA3-LRWHM and AES-SHA3-RHM respectively. According to the

[10] As our goal is to benchmark and demonstrate a comparison to AES-CBC, we didn't adopt various optimization techniques such as inline assembly programming.

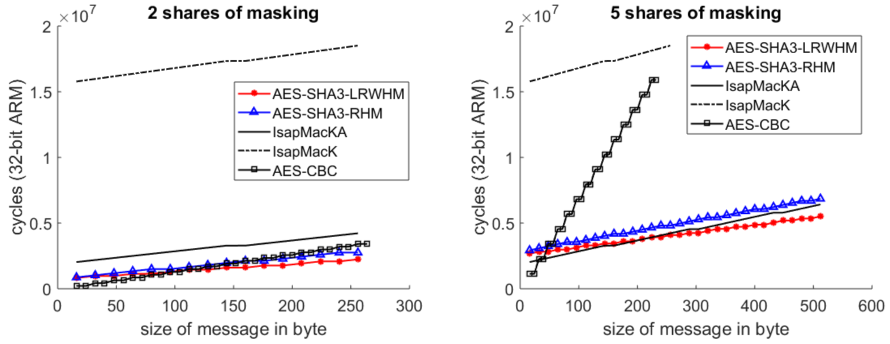

Figure 6: Performance (I): the consumed clock cycles of AES-SHA3-LRWHM, AES-SHA3-RHM, IsapMacs and AES-CBC with varying the size of messages (lower is better).

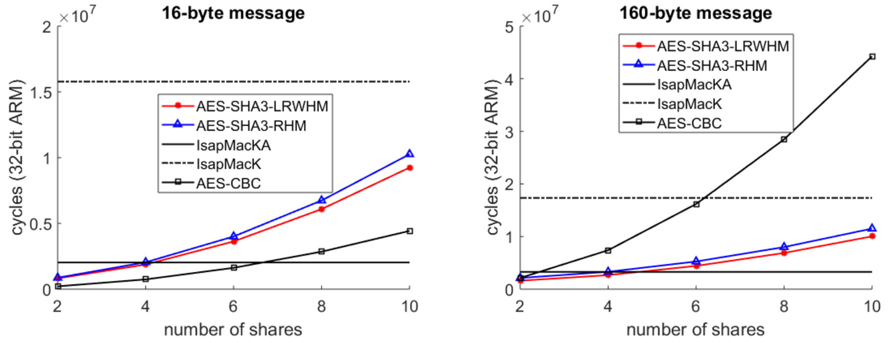

Figure 7: Performance (II): the clock cycles of AES-SHA3-LRWHM, AES-SHA3-RHM, IsapMacs and AES-CBC with varying the number of shares (lower is better).

$c/2$ indifferentiable bound [BDPV08], for AES-SHA3-LRWHM we set $r = 224$, so that $c/2 \geq 78.3$ won't constitute the security bottleneck; similarly, for AES-SHA3-RHM we set $r = 160$ so that $c/2 = 120$. As mentioned in the Introduction, once deployed, their main threat should be the side-channel key recovery against the AES implementations.

For the sake of efficiency, we pre-expand the MAC keys and store them in memory under a shared form. That is, for AES-SHA3-LRWHM we never execute the key schedule (since it only calls AES with fixed keys), and for AES-SHA3-RHM, we only need to execute the (masked) key schedule once (due to the rekeying of the second AES-call). Based on all these, we benchmark the performance of our schemes on the 32-bit ARM Cortex-M3 processor, as depicted in Fig. 6 and 7. Unsurprisingly, AES-SHA3-RHM is slightly more costly than AES-SHA3-LRWHM due to the additional (masked) AES key schedule.

For comparison, we also consider the IsapMacK and IsapMacKA implementations in [DEM+19], the MAC functions underlying their Keccak-based AE variants Isap-K-128 and Isap-K-128A resp. The main difference between Isap-K-128 and Isap-K-128A lies in the rate-1 duplex-based FILTG function: Isap-K-128 invokes 12-round Keccak-f[400] for more reliable security, while Isap-K-128A invokes 1-round Keccak-f[400] for better efficiency [DEM+19]. We also implement a simplified AES-CBC variant built upon our masked AES with pre-expanded secret keys, in order to obtain the performance *upper bound* of CBC (see Appendix C for the pseudocode of this CBC variant). Note that here we focus on *performance comparison*, and thus we don't care about their (different) security bounds. To ease understanding, we picture all the evaluation results in Figs. 6 and 7. Our source code has been submitted as the separate supplementary material (an zip archive).

Among them, Fig 6 illustrates the impacts of message size on the latency (in terms of the number of cycles for processing), in which the X axis represents the message size, and the two sub-figures depict clock cycles of AES-SHA3-LRWHM, AES-SHA3-RHM, IsapMacK, IsapMacKA and AES-CBC on the Y axis with number of shares $d = 2$ and 5 respectively for masked AES. The first observation is that AES-SHA3-LRWHM and AES-SHA3-RHM outperform IsapMacK when up to 5 shares are used, and are comparable to IsapMacKA when $d = 5$ (when $d = 2$, IsapMacKA is slightly inferior). Such gains stem from the relatively low performance of the rate-1 duplex in IsapMacK and IsapMacKA. Though, as mentioned, the rate-1 duplex in IsapMacKA is extremely light, and thus ours don't have much advantage. Another interesting observation is that the latency of AES-CBC greatly increases with the message size—even forming "stairs", which is in sharp contrast to the smooth curves of our algorithms (and the two IsapMacs): this is because in our algorithms (and IsapMacs), every additional message block only induces (roughly) a call to the efficient (unprotected) Keccak-$f[400]$ permutation, the cost of which is negligible compared to the one more masked AES-call in AES-CBC. Due to this, our algorithms outperform AES-CBC as long as the message contains more than 120 and 50 bytes for masking order $d = 2$ and $d = 5$ resp., and the gains further increase with the size.

On the other hand, Fig 7 takes the X axis for the number of shares and reflects the impacts of side-channel protection (in terms of the number of shares) on the latency. The left sub-figure shows cycles of aforementioned MACs on the Y axis when processing messages with only 16 bytes, while the right shows those for longer messages with 160 bytes. It's thus natural to see that the performance gains of AES-SHA3-LRWHM, AES-SHA3-RHM over IsapMacs decrease with the masking order d, since IsapMacs rely on the masking order-independent rate-1 duplex. Still, with less than 10-share masked AES (which already corresponds to a significantly higher security level than actually deployed), our schemes are more efficient than IsapMacK; with less than 4-share masked AES, our schemes outperform IsapMacKA. While Fig 7 (left) seems to indicate AES-CBC is better, we stress that the comparison is made w.r.t. very short messages of only a single block. For such short inputs, our deficiency is expected, since our schemes make 2 AES-calls while the CBC variant makes only 1 AES-call. But as long as the message turns longer, e.g., 160 bytes in Fig 7 (right), our algorithms achieve performance gains that increase significantly with the number of shares.

In summary, AES-SHA3-LRWHM and AES-SHA3-RHM outperform AES-CBC as long as the message consists of more than 120 and 50 bytes for masking order $d = 2$ and $d = 5$ respectively, and the performance gains increase with both the message size and the strength of side-channel protection. They also outperform the IsapMacK and IsapMacKA implementations with up to probing secure orders 9 and 3 respectively (corresponding to 10-share and 4-share masked AES). We remark again that the goals of our (masked) schemes and IsapMacs are quite different—as discussed in the Introduction.

Note that we took AES-CBC as a representative of the "fully protected" classical MACs, and omit the other such as HMAC, KMAC, Wegman-Carter, and ZMAC [IMPS17]. As discussed in the Introduction, all of them consume $\geq \ell$ heavy protected executions (of permutations, field multiplications, etc), which is similar to AES-CBC. Therefore, their performances are expected to be similar to AES-CBC, i.e., the latency due to the masked primitives increases linearly with the message size. Also, compared to AES, a more significant performance loss is expected from protecting SHA2, due to the relatively high complexity (generally $\mathcal{O}(d^2 \log k)$ for register size k) of higher-order conversion from Boolean to arithmetic masking [CGTV15, CGV14]. For MACs using multiplication-based universal hash functions, the latency could be decreased via parallel implementations (though may be a bit difficult for the ARM settings), but the energy consumption remains remarkable. In summary, for the protected standards, the performance of AES-CBC is expected to be among the best, thus constituting a reasonable baseline.

5 Concluding Remarks

We propose two MA modes LRWHM and RHM that for the first time achieve provable beyond-birthday security when the protected blockciphers are leakage secure, but most other intermediate values during tag and verification computations are leaked. The modes can be easily deployed. We benchmark performances for their instances, which exhibits advantages over existing schemes or standards.

Acknowledgments

We thank Itamar Levi for fruitful discussion. Chun Guo was partly supported by the Program of Qilu Young Scholars (Grant No. 61580089963177) of Shandong University. François-Xavier Standaert is a senior research associate of the Belgian Fund for Scientific Research (F.R.S.-FNRS). This work has been funded in parts by the European Union through the ERC project SWORD (724725), the INNOVIRIS projects SCAUT and C-Cure, and the European Union and Walloon Region FEDER USERMedia project 501907-379156. Weijia Wang was partly supported by the Program of Qilu Young Scholars of Shandong University. Yu Yu is supported by the National Natural Science Foundation of China (Grant Nos. 61872236, 61472249, 61572192) and the National Cryptography Development Fund MMJJ20170209.

References

[ABD+13] Elena Andreeva, Andrey Bogdanov, Yevgeniy Dodis, Bart Mennink, and John P. Steinberger. On the indifferentiability of key-alternating ciphers. In Ran Canetti and Juan A. Garay, editors, *CRYPTO 2013, Part I*, volume 8042 of *LNCS*, pages 531–550. Springer, Heidelberg, August 2013.

[BCK96] Mihir Bellare, Ran Canetti, and Hugo Krawczyk. Keying hash functions for message authentication. In Neal Koblitz, editor, *CRYPTO'96*, volume 1109 of *LNCS*, pages 1–15. Springer, Heidelberg, August 1996.

[BDPV08] Guido Bertoni, Joan Daemen, Michaël Peeters, and Gilles Van Assche. On the indifferentiability of the sponge construction. In Nigel P. Smart, editor, *EUROCRYPT 2008*, volume 4965 of *LNCS*, pages 181–197. Springer, Heidelberg, April 2008.

[BGM04] Mihir Bellare, Oded Goldreich, and Anton Mityagin. The Power of Verification Queries in Message Authentication and Authenticated Encryption. *IACR Cryptology ePrint Archive*, 2004:309, 2004.

[BGP+19] Francesco Berti, Chun Guo, Olivier Pereira, Thomas Peters, and François-Xavier Standaert. TEDT, a Leakage-Resilient AEAD mode for High (Physical) Security Applications. Cryptology ePrint Archive, Report 2019/137, 2019. https://eprint.iacr.org/2019/137. To appear at CHES 2020.

[BKP+18] Francesco Berti, François Koeune, Olivier Pereira, Thomas Peters, and François-Xavier Standaert. Ciphertext integrity with misuse and leakage: Definition and efficient constructions with symmetric primitives. In Jong Kim, Gail-Joon Ahn, Seungjoo Kim, Yongdae Kim, Javier López, and Taesoo Kim, editors, *ASIACCS 18*, pages 37–50. ACM Press, April 2018.

[BKR98] Mihir Bellare, Ted Krovetz, and Phillip Rogaway. Luby-Rackoff backwards: Increasing security by making block ciphers non-invertible. In Kaisa Nyberg, editor, *EUROCRYPT'98*, volume 1403 of *LNCS*, pages 266–280. Springer, Heidelberg, May / June 1998.

[BKR00] Mihir Bellare, Joe Kilian, and Phillip Rogaway. The Security of the Cipher Block Chaining Message Authentication Code. *J. Comput. Syst. Sci.*, 61(3):362–399, 2000.

[BMOS17] Guy Barwell, Daniel P. Martin, Elisabeth Oswald, and Martijn Stam. Authenticated encryption in the face of protocol and side channel leakage. In Tsuyoshi Takagi and Thomas Peyrin, editors, *ASIACRYPT 2017, Part I*, volume 10624 of *LNCS*, pages 693–723. Springer, Heidelberg, December 2017.

[BN08] Mihir Bellare and Chanathip Namprempre. Authenticated encryption: Relations among notions and analysis of the generic composition paradigm. *Journal of Cryptology*, 21(4):469–491, October 2008.

[BPPS17] Francesco Berti, Olivier Pereira, Thomas Peters, and François-Xavier Standaert. On leakage-resilient authenticated encryption with decryption leakages. *IACR Trans. Symm. Cryptol.*, 2017(3):271–293, 2017.

[BR02] John Black and Phillip Rogaway. A block-cipher mode of operation for parallelizable message authentication. In Lars R. Knudsen, editor, *EUROCRYPT 2002*, volume 2332 of *LNCS*, pages 384–397. Springer, Heidelberg, April / May 2002.

[BR06] Mihir Bellare and Phillip Rogaway. The security of triple encryption and a framework for code-based game-playing proofs. In Serge Vaudenay, editor, *EUROCRYPT 2006*, volume 4004 of *LNCS*, pages 409–426. Springer, Heidelberg, May / June 2006.

[CGH98] Ran Canetti, Oded Goldreich, and Shai Halevi. The random oracle methodology, revisited (preliminary version). In *30th ACM STOC*, pages 209–218. ACM Press, May 1998.

[CGTV15] Jean-Sébastien Coron, Johann Großschädl, Mehdi Tibouchi, and Praveen Kumar Vadnala. Conversion from arithmetic to Boolean masking with logarithmic complexity. In Gregor Leander, editor, *FSE 2015*, volume 9054 of *LNCS*, pages 130–149. Springer, Heidelberg, March 2015.

[CGV14] Jean-Sébastien Coron, Johann Großschädl, and Praveen Kumar Vadnala. Secure conversion between Boolean and arithmetic masking of any order. In Lejla Batina and Matthew Robshaw, editors, *CHES 2014*, volume 8731 of *LNCS*, pages 188–205. Springer, Heidelberg, September 2014.

[CLL$^+$18] Shan Chen, Rodolphe Lampe, Jooyoung Lee, Yannick Seurin, and John P. Steinberger. Minimizing the two-round Even-Mansour cipher. *Journal of Cryptology*, 31(4):1064–1119, October 2018.

[CLS17] Benoît Cogliati, Jooyoung Lee, and Yannick Seurin. New constructions of macs from (tweakable) block ciphers. *IACR Trans. Symm. Cryptol.*, 2017(2):27–58, 2017.

[CRR03] Suresh Chari, Josyula R. Rao, and Pankaj Rohatgi. Template attacks. In Burton S. Kaliski Jr., Çetin Kaya Koç, and Christof Paar, editors, *CHES 2002*, volume 2523 of *LNCS*, pages 13–28. Springer, Heidelberg, August 2003.

[CS16] Benoît Cogliati and Yannick Seurin. EWCDM: An efficient, beyond-birthday secure, nonce-misuse resistant MAC. In Matthew Robshaw and Jonathan Katz, editors, *CRYPTO 2016, Part I*, volume 9814 of *LNCS*, pages 121–149. Springer, Heidelberg, August 2016.

[DDN+17] Nilanjan Datta, Avijit Dutta, Mridul Nandi, Goutam Paul, and Liting Zhang. Single key variant of PMAC_Plus. *IACR Trans. Symm. Cryptol.*, 2017(4):268–305, 2017.

[DDNP18] Nilanjan Datta, Avijit Dutta, Mridul Nandi, and Goutam Paul. Double-block hash-then-sum: A paradigm for constructing BBB secure PRF. *IACR Trans. Symm. Cryptol.*, 2018(3):36–92, 2018.

[DDNY18] Nilanjan Datta, Avijit Dutta, Mridul Nandi, and Kan Yasuda. Encrypt or decrypt? To make a single-key beyond birthday secure nonce-based MAC. In Hovav Shacham and Alexandra Boldyreva, editors, *CRYPTO 2018, Part I*, volume 10991 of *LNCS*, pages 631–661. Springer, Heidelberg, August 2018.

[DEM+17] Christoph Dobraunig, Maria Eichlseder, Stefan Mangard, Florian Mendel, and Thomas Unterluggauer. ISAP – towards side-channel secure authenticated encryption. *IACR Trans. Symm. Cryptol.*, 2017(1):80–105, 2017.

[DEM+19] Christoph Dobraunig, Maria Eichlseder, Stefan Mangard, Florian Mendel, Bart Mennink, and Robert Primas. ISAP v2.0. Submission to NIST, 2019. Specification: https://csrc.nist.gov/CSRC/media/Projects/Lightweight-Cryptography/documents/round-1/spec-doc/ISAP-spec.pdf. Code: https://csrc.nist.gov/CSRC/media/Projects/Lightweight-Cryptography/documents/round-1/submissions/isap.zip.

[DM19] Christoph Dobraunig and Bart Mennink. Leakage Resilience of the Duplex Construction. Cryptology ePrint Archive, Report 2019/225, 2019. https://eprint.iacr.org/2019/225.

[DP08] Stefan Dziembowski and Krzysztof Pietrzak. Leakage-resilient cryptography. In *49th FOCS*, pages 293–302. IEEE Computer Society Press, October 2008.

[DS09] Yevgeniy Dodis and John P. Steinberger. Message authentication codes from unpredictable block ciphers. In Shai Halevi, editor, *CRYPTO 2009*, volume 5677 of *LNCS*, pages 267–285. Springer, Heidelberg, August 2009.

[DS11] Yevgeniy Dodis and John P. Steinberger. Domain extension for MACs beyond the birthday barrier. In Kenneth G. Paterson, editor, *EUROCRYPT 2011*, volume 6632 of *LNCS*, pages 323–342. Springer, Heidelberg, May 2011.

[GLSV15] Vincent Grosso, Gaëtan Leurent, François-Xavier Standaert, and Kerem Varici. LS-designs: Bitslice encryption for efficient masked software implementations. In Carlos Cid and Christian Rechberger, editors, *FSE 2014*, volume 8540 of *LNCS*, pages 18–37. Springer, Heidelberg, March 2015.

[GMK17] Hannes Groß, Stefan Mangard, and Thomas Korak. An efficient side-channel protected AES implementation with arbitrary protection order. In Helena Handschuh, editor, *CT-RSA 2017*, volume 10159 of *LNCS*, pages 95–112. Springer, Heidelberg, February 2017.

[GPPS] Chun Guo, Olivier Pereira, Thomas Peters, and François-Xavier Standaert. Authenticated Encryption with Nonce Misuse and Physical Leakages: Definitions, Separation Results, and Leveled Constructions. Cryptology ePrint Archive, Report 2018/484. Appeared at LATINCRYPT 2019.

[GPPS19] Chun Guo, Olivier Pereira, Thomas Peters, and François-Xavier Standaert. Towards Low-Energy Leakage-Resistant Authenticated Encryption from the Duplex Sponge Construction. Cryptology ePrint Archive, Report 2019/193, 2019. https://eprint.iacr.org/2019/193.

[GR17] Dahmun Goudarzi and Matthieu Rivain. How fast can higher-order masking be in software? In Jean-Sébastien Coron and Jesper Buus Nielsen, editors, *EUROCRYPT 2017, Part I*, volume 10210 of *LNCS*, pages 567–597. Springer, Heidelberg, April / May 2017.

[HLWW16] Carmit Hazay, Adriana López-Alt, Hoeteck Wee, and Daniel Wichs. Leakage-resilient cryptography from minimal assumptions. *Journal of Cryptology*, 29(3):514–551, July 2016.

[IMPS17] Tetsu Iwata, Kazuhiko Minematsu, Thomas Peyrin, and Yannick Seurin. ZMAC: A fast tweakable block cipher mode for highly secure message authentication. In Jonathan Katz and Hovav Shacham, editors, *CRYPTO 2017, Part III*, volume 10403 of *LNCS*, pages 34–65. Springer, Heidelberg, August 2017.

[KCP16] John Kelsey, Shu-jen Chang, and Ray Perlner. Sha-3 derived functions: cshake, kmac, tuplehash, and parallelhash. Technical report, National Institute of Standards and Technology, 2016.

[KJJ99] Paul C. Kocher, Joshua Jaffe, and Benjamin Jun. Differential power analysis. In Michael J. Wiener, editor, *CRYPTO'99*, volume 1666 of *LNCS*, pages 388–397. Springer, Heidelberg, August 1999.

[Koc96] Paul C. Kocher. Timing attacks on implementations of Diffie-Hellman, RSA, DSS, and other systems. In Neal Koblitz, editor, *CRYPTO'96*, volume 1109 of *LNCS*, pages 104–113. Springer, Heidelberg, August 1996.

[LN17] Eik List and Mridul Nandi. ZMAC$^+$ – an efficient variable-output-length variant of ZMAC. *IACR Trans. Symm. Cryptol.*, 2017(4):306–325, 2017.

[LRW11] Moses Liskov, Ronald L. Rivest, and David Wagner. Tweakable block ciphers. *Journal of Cryptology*, 24(3):588–613, July 2011.

[LST12] Will Landecker, Thomas Shrimpton, and R. Seth Terashima. Tweakable blockciphers with beyond birthday-bound security. In Reihaneh Safavi-Naini and Ran Canetti, editors, *CRYPTO 2012*, volume 7417 of *LNCS*, pages 14–30. Springer, Heidelberg, August 2012.

[Men17] Bart Mennink. Insuperability of the standard versus ideal model gap for tweakable blockcipher security. In Jonathan Katz and Hovav Shacham, editors, *CRYPTO 2017, Part II*, volume 10402 of *LNCS*, pages 708–732. Springer, Heidelberg, August 2017.

[Min09] Kazuhiko Minematsu. Beyond-birthday-bound security based on tweakable block cipher. In Orr Dunkelman, editor, *FSE 2009*, volume 5665 of *LNCS*, pages 308–326. Springer, Heidelberg, February 2009.

[MN17] Bart Mennink and Samuel Neves. Encrypted Davies-Meyer and its dual: Towards optimal security using mirror theory. In Jonathan Katz and Hovav Shacham, editors, *CRYPTO 2017, Part III*, volume 10403 of *LNCS*, pages 556–583. Springer, Heidelberg, August 2017.

[MOP07] Stefan Mangard, Elisabeth Oswald, and Thomas Popp. *Power Analysis Attacks - Revealing the Secrets of Smart Cards*. Springer, 2007.

[MOSW15] Daniel P. Martin, Elisabeth Oswald, Martijn Stam, and Marcin Wójcik. A leakage resilient MAC. In Jens Groth, editor, *15th IMA International Conference on Cryptography and Coding*, volume 9496 of *LNCS*, pages 295–310. Springer, Heidelberg, December 2015.

[MRH04] Ueli M. Maurer, Renato Renner, and Clemens Holenstein. Indifferentiability, impossibility results on reductions, and applications to the random oracle methodology. In Moni Naor, editor, *TCC 2004*, volume 2951 of *LNCS*, pages 21–39. Springer, Heidelberg, February 2004.

[oSN18] National Institute of Standards and Technology (NIST). DRAFT Submission Requirements and Evaluation Criteria for the Lightweight Cryptography Standardization Process. NIST official website, 2018. https://csrc.nist.gov/CSRC/media/Projects/Lightweight-Cryptography/documents/Draft-LWC-Submission-Requirements-April2018.pdf.

[Pie09] Krzysztof Pietrzak. A leakage-resilient mode of operation. In Antoine Joux, editor, *EUROCRYPT 2009*, volume 5479 of *LNCS*, pages 462–482. Springer, Heidelberg, April 2009.

[PR00] Erez Petrank and Charles Rackoff. CBC MAC for real-time data sources. *Journal of Cryptology*, 13(3):315–338, June 2000.

[PS15] Thomas Peyrin and Yannick Seurin. Counter-in-Tweak: Authenticated Encryption Modes for Tweakable Block Ciphers. Version 20160524:153228. Cryptology ePrint Archive, Report 2015/1049, 2015. Extended abstract appeared at CRYPTO 2016.

[PSP$^+$08] Christophe Petit, François-Xavier Standaert, Olivier Pereira, Tal Malkin, and Moti Yung. A block cipher based pseudo random number generator secure against side-channel key recovery. In Masayuki Abe and Virgil Gligor, editors, *ASIACCS 08*, pages 56–65. ACM Press, March 2008.

[PSV15] Olivier Pereira, François-Xavier Standaert, and Srinivas Vivek. Leakage-resilient authentication and encryption from symmetric cryptographic primitives. In Indrajit Ray, Ninghui Li, and Christopher Kruegel, editors, *ACM CCS 2015*, pages 96–108. ACM Press, October 2015.

[RKS17] Usman Raza, Parag Kulkarni, and Mahesh Sooriyabandara. Low Power Wide Area Networks: An Overview. *IEEE Communications Surveys and Tutorials*, 19(2):855–873, 2017.

[Rog04] Phillip Rogaway. Efficient instantiations of tweakable blockciphers and refinements to modes OCB and PMAC. In Pil Joong Lee, editor, *ASIACRYPT 2004*, volume 3329 of *LNCS*, pages 16–31. Springer, Heidelberg, December 2004.

[RSS17] Dragos Rotaru, Nigel P. Smart, and Martijn Stam. Modes of operation suitable for computing on encrypted data. *IACR Trans. Symm. Cryptol.*, 2017(3):294–324, 2017.

[RSV09] Mathieu Renauld, François-Xavier Standaert, and Nicolas Veyrat-Charvillon. Algebraic side-channel attacks on the AES: why time also matters in DPA. In Christophe Clavier and Kris Gaj, editors, *Cryptographic Hardware and Embedded Systems - CHES 2009, 11th International Workshop, Lausanne,* *Switzerland, September 6-9, 2009, Proceedings*, volume 5747 of *Lecture Notes in Computer Science*, pages 97–111. Springer, 2009.

[RSWO17] Eyal Ronen, Adi Shamir, Achi-Or Weingarten, and Colin O'Flynn. IoT goes nuclear: Creating a ZigBee chain reaction. In *2017 IEEE Symposium on Security and Privacy*, pages 195–212. IEEE Computer Society Press, May 2017.

[SBK+17] Marc Stevens, Elie Bursztein, Pierre Karpman, Ange Albertini, and Yarik Markov. The first collision for full SHA-1. In Jonathan Katz and Hovav Shacham, editors, *CRYPTO 2017, Part I*, volume 10401 of *LNCS*, pages 570–596. Springer, Heidelberg, August 2017.

[Sch10] J. Schipper. Leakage-Resilient Authentication. Ph.D. thesis, Utrecht University, 2010.

[SPWS13] Laszlo Szekeres, Mathias Payer, Tao Wei, and Dawn Song. SoK: Eternal war in memory. In *2013 IEEE Symposium on Security and Privacy*, pages 48–62. IEEE Computer Society Press, May 2013.

[SPY13] François-Xavier Standaert, Olivier Pereira, and Yu Yu. Leakage-resilient symmetric cryptography under empirically verifiable assumptions. In Ran Canetti and Juan A. Garay, editors, *CRYPTO 2013, Part I*, volume 8042 of *LNCS*, pages 335–352. Springer, Heidelberg, August 2013.

[ST16] Thomas Shrimpton and R. Seth Terashima. Salvaging weak security bounds for blockcipher-based constructions. In Jung Hee Cheon and Tsuyoshi Takagi, editors, *ASIACRYPT 2016, Part I*, volume 10031 of *LNCS*, pages 429–454. Springer, Heidelberg, December 2016.

[VGS14] Nicolas Veyrat-Charvillon, Benoît Gérard, and François-Xavier Standaert. Soft analytical side-channel attacks. In Palash Sarkar and Tetsu Iwata, editors, *Advances in Cryptology - ASIACRYPT 2014 - 20th International Conference on the Theory and Application of Cryptology and Information Security, Kaoshiung, Taiwan, R.O.C., December 7-11, 2014. Proceedings, Part I*, volume 8873 of *Lecture Notes in Computer Science*, pages 282–296. Springer, 2014.

[WC81] Mark N. Wegman and Larry Carter. New Hash Functions and Their Use in Authentication and Set Equality. *J. Comput. Syst. Sci.*, 22(3):265–279, 1981.

[WHF] D. Whiting, R. Housley, and N. Ferguson. Counter with CBC-MAC (CCM). RFC 3610, Oct. 2015. Available: https://rfc-editor.org/rfc/rfc3610.txt.

[WYS+18] Weijia Wang, Yu Yu, François-Xavier Standaert, Junrong Liu, Zheng Guo, and Dawu Gu. Ridge-Based DPA: Improvement of Differential Power Analysis For Nanoscale Chips. *IEEE Trans. Information Forensics and Security*, 13(5):1301–1316, 2018.

[Yas09] Kan Yasuda. A double-piped mode of operation for MACs, PRFs and PROs: Security beyond the birthday barrier. In Antoine Joux, editor, *EUROCRYPT 2009*, volume 5479 of *LNCS*, pages 242–259. Springer, Heidelberg, April 2009.

[Yas11] Kan Yasuda. A new variant of PMAC: Beyond the birthday bound. In Phillip Rogaway, editor, *CRYPTO 2011*, volume 6841 of *LNCS*, pages 596–609. Springer, Heidelberg, August 2011.

[YSPY10] Yu Yu, François-Xavier Standaert, Olivier Pereira, and Moti Yung. Practical leakage-resilient pseudorandom generators. In Ehab Al-Shaer, Angelos D. Keromytis, and Vitaly Shmatikov, editors, *ACM CCS 2010*, pages 141–151. ACM Press, October 2010.

[ZBPB17] Jean Karim Zinzindohoué, Karthikeyan Bhargavan, Jonathan Protzenko, and Benjamin Beurdouche. HACL*: A verified modern cryptographic library. In Bhavani M. Thuraisingham, David Evans, Tal Malkin, and Dongyan Xu, editors, *ACM CCS 2017*, pages 1789–1806. ACM Press, October / November 2017.

[ZS18] Mark Zhao and G. Edward Suh. FPGA-based remote power side-channel attacks. In *2018 IEEE Symposium on Security and Privacy*, pages 229–244. IEEE Computer Society Press, May 2018.

[ZWSW12] Liting Zhang, Wenling Wu, Han Sui, and Peng Wang. 3kf9: Enhancing 3GPP-MAC beyond the birthday bound. In Xiaoyun Wang and Kazue Sako, editors, *ASIACRYPT 2012*, volume 7658 of *LNCS*, pages 296–312. Springer, Heidelberg, December 2012.

A Justifying Eq. (14)

For any V, we follow Chen et al. [CLL+18] to eliminate the dependency between $L_1(V), \ldots, L_w(V)$ and then apply the multiplicative Chernoff bound. In detail, consider a sequence of independent and identically distributed random variables $B_1, \ldots, B_w$ such that

$$\Pr[B_i = 0] = 1 - \frac{2q}{2^n} \quad \text{and} \quad \Pr[B_i = 1] = \frac{2q}{2^n}.$$

Let $\mu = (1+\delta) \cdot \frac{2wq}{2^n}$. We prove that

$$\Pr\Big[\sum_{i=1}^{w} L_i(V) \geq \mu\Big] \leq \Pr\Big[\sum_{i=1}^{w} B_i \geq \mu\Big]. \tag{20}$$

The coupling-like argument follows the proof of Lemma 5 in [CLL+18]. Let Ber_p denote the Bernoulli distribution of parameter p. Consider the following sampling process:

```
for $i = 1$ to $w$ do
  $p := \Pr[L_i(V) = 1 \mid (L_1(V), \ldots, L_{i-1}(V)) = (a_1, \ldots, a_{i-1})]$
  $u_i \leftarrow \mathsf{Ber}_p$
  if $u_i = 1$ then $v_i := 1$
  else
    $p' := \frac{\frac{2q}{2^n} - p}{1-p}$, $v_i \leftarrow \mathsf{Ber}_{p'}$
return $((u_1, \ldots, u_w), (v_1, \ldots, v_w))$
```

Then clearly $(u_1, \ldots, u_w)$ follows the distribution of $L_1(V), \ldots, L_w(V)$. Moreover, v_i follows the distribution of B_i for any $i = 1, \ldots, w$: in fact, for any i and any sequence $(v_1, \ldots, v_{i-1}) \in \{0,1\}^{i-1}$, it holds

$$Pr[v_i = 1 \mid (v_1, \ldots, v_{i-1})] = p + p'(1-p) = \frac{2q}{2^n}.$$

During the sampling process, $u_i = 1$ implies $v_i = 1$, so that for any r,

$$\sum_{i=1}^{w} u_i \geq r \Rightarrow \sum_{i=1}^{w} v_i \geq r.$$

This implies Eq. (20).

For the r.h.s. of Eq. (20), since $\mathbb{E}\big[\sum_{i=1}^{w} B_i\big] = \frac{2wq}{2^n}$, the multiplicative Chernoff bound states that for any $\delta > 0$,

$$\Pr\Big[\sum_{i=1}^{w} B_i \geq (1+\delta)\mu\Big] \leq \left(\frac{e^{\delta}}{(1+\delta)^{1+\delta}}\right)^{\frac{2wq}{2^n}} < \left(\frac{e}{1+\delta}\right)^{\mu}.$$

To further simplify, let $\theta = 2wq$. The remaining arguments are almost the same as Appendix A of [PS15]: when $\theta = 2wq < 8$, by the assumption $q \geq 4$ it has to be $w = 0$. Therefore, it trivially holds

$$\Pr\Big[\exists V \in \{0,1\}^n : \sum_{i=1}^{0} L_i(V) \geq 8\sqrt{q} + \frac{8nq^2}{2^n}\Big] = 0 \leq \frac{1}{2^n}.$$

When $8 \leq \theta \leq 2^n$, taking $\delta = \frac{2^{n+1}\log\theta}{\theta} - 1$ for logarithms in base 2 yields

$$\begin{aligned}\Pr\big[\sum_{i=1}^{w} B_i \geq 2\log\theta\big] &\leq \left(\frac{e\theta}{2^{n+1}\log\theta}\right)^{2\log\theta} \\ &= \frac{\theta^2}{2^{2n}}\left(\frac{e}{2\log\theta}\right)^{2\log\theta}\left(\frac{\theta}{2^n}\right)^{2\log\theta-2} \\ &\leq \frac{\theta^2}{2^{2n}}\left(\frac{1}{2}\right)^{2\log\theta}\left(\frac{\theta}{2^n}\right)^{2\log\theta-2} \qquad (\theta \geq 8) \\ &= \frac{1}{2^{2n}}\left(\frac{\theta}{2^n}\right)^{2\log\theta-2} \leq \frac{1}{2^{2n}}, \qquad (\theta \leq 2^n).\end{aligned}$$

Therefore,

$$\Pr\Big[\exists V \in \{0,1\}^n : \sum_{i=1}^{w} L_i(V) \geq 2\log\theta\Big] \leq 2^n \cdot \frac{1}{2^{2n}} \leq \frac{1}{2^n}.$$

When $\theta \geq 2^n$, taking $\delta = 2n - 1$ yields

$$\begin{aligned}\Pr\Big[\sum_{i=1}^{w} B_i \geq \frac{2n\theta}{2^n}\Big] &\leq \left(\frac{e}{2n}\right)^{\frac{2n\theta}{2^n}} \leq \left(\frac{1}{2}\right)^{\frac{2n\theta}{2^n}} \qquad (2^n \geq 8) \\ &\leq \left(\frac{1}{2}\right)^{2n} \leq \frac{1}{2^{2n}}. \qquad (\theta \geq 2^n)\end{aligned}$$

Again, this means

$$\Pr\Big[\exists V \in \{0,1\}^n : \sum_{i=1}^{w} L_i(V) \geq \frac{2n\theta}{2^n}\Big] \leq \frac{1}{2^n}.$$

Note that $w \leq 2q$, which means $2\log\theta + \frac{2n\theta}{2^n} \leq 4 + 4\log q + \frac{8nq^2}{2^n} \leq 8\sqrt{q} + \frac{8nq^2}{2^n}$. By this and all the above, in any case,

$$\Pr\Big[\exists V \in \{0,1\}^n : \sum_{i=1}^{w} L_i(V) \geq 8\sqrt{q} + \frac{8nq^2}{2^n}\Big] \leq \frac{1}{2^n}.$$

B Tightness of Theorem 1

In the unbounded leakage model, we have a matching attack on LRWHM as follow:

(1) For $q = 2^{2n/3}$, arbitrarily choose $2q$ distinct messages $M_1, \ldots, M_q; M'_1, \ldots, M'_q$;

(2) Make q tag queries, i.e. $\mathsf{LTGen}(M_1) \to T_1$, ..., $\mathsf{LTGen}(M_q) \to T_q$, and collect $2q$ corresponding pairs $\mathsf{E}_{K_1}(U_1) = V_1, \ldots, \mathsf{E}_{K_1}(U_q) = V_q$; $\mathsf{E}_{K_2}(Y_1) = T_1, \ldots, \mathsf{E}_{K_2}(Y_q) = T_q$ from the (unbounded) leakages;

(3) Make q random oracle queries for the unused messages, i.e. $\mathsf{RO}(M'_1) \to U'_1X'_1$, ..., $\mathsf{RO}(M'_q) \to U'_qX'_q$;

(4) Seek for a triple of records $\mathsf{RO}(M'_i) = U'_iX'_i, \mathsf{E}_{K_1}(U_j) = V_j, \mathsf{E}_{K_2}(Y_\ell) = T_\ell$ such that $U'_i = U_j$ and $X'_i = V_j \oplus Y_\ell$, and output (M'_i, T_ℓ) as a forgery.

Since U_i is uniform for any i, it holds

$$\Pr\Big[\big|\{(M_i, U_iX_i) : \exists (M_j, U_iX_j), j \neq i\}\big| \geq \sqrt{q}\Big] \leq \frac{q^{3/2}}{N}.$$

This means with probability greater than 1, the number of distinct tuples $\mathsf{E}_{K_1}(U_i) = V_i$ obtained during step 2 is $\geq q - \sqrt{q} = O(q)$. Similarly, since X_i is uniform for any i, the number of distinct tuples $\mathsf{E}_{K_2}(Y_i) = T_i$ obtained during step 2 is $O(q)$. This means the number of observed distinct pairs $(\mathsf{E}_{K_1}(U_j) = V_j, \mathsf{E}_{K_2}(Y_\ell) = T_\ell)$ is $O(q^2)$. By these, the probability of forgery is a significant constant. We eschew the detailed calculations. It's not hard to see the goal of this attack is exactly to trigger the previous condition (C-1).

However, note that the data complexity of the above attack is $2^{2n/3}$. Thus it's unlikely more efficient than a naïve side-channel key recovery. We are not aware of any attack simultaneously achieving low(er) data and comparable $2^{2n/3}$ time complexities (low data attacks with much heavier computations, e.g., 2^n, do exist, but they are impractical).

C Pseudocode for the CBC Variant in Section 4

The tag generation $\mathsf{TGen}_K^{\mathsf{E}}(M)$ of the CBC variant is described as follows.

```
1. Pads $M$ and parses it into $n$-bit blocks $M = M_1 \| \ldots \| M_\ell \| 0^*$, such that $|M_1| = \ldots = |M_\ell \| 0^*| = n$, and the number of padded 0 is minimal.
2. $T := IV$        // $IV$ is typically a fixed constant
3. for $i = 1$ to $\ell$ do
     • $T := T \oplus M_i$
     • $T := \mathsf{E}_K(T)$
4. return $T$
```

We note that the above padding isn't secure. But as our goal is to estimate the *efficiency upper bound* for various CBC variants, this variant is sufficient. Indeed, it provides a *lower bound* on the number of AES-calls: some variants such as EMAC [PR00] make additional calls.